

ادخل إلى عالم

ORACLE

المهندس مهيب النقري




ادخل إلى عالم
Oracle ٨
إعداد
المهندس مهيب النقري

عناوين صدرت في سلسلة الرضا للمعلومات

اسم الكتاب	المؤلف	تاريخ النشر
١- بيئة النوافذ WINDOWS 3.11	م. أحمد شربك	١٩٩٤
٢- مبادىء الصيانة والشبكات	م. عبد الله أحمد	١٩٩٤
٣- معالجة النصوص MS WORD 6.0	د. هيثم البيطار	١٩٩٥
٤- نظام تشغيل WINDOWS 95	م. مهيب النقري	١٩٩٦
٥- قواعد البيانات MS ACCESS	زياد كمرجي - بيداء الزير	١٩٩٧
٦- توابع وماكروات في MS EXCEL 97	أ. زياد كمرجي	١٩٩٧
٧- مرجع تعليمي شامل لبرنامج		
معالجة النصوص MS WORD 97	د. هيثم البيطار	١٩٩٧
٨- مرجع تعليمي شامل في MS EXCEL 97	أ. زياد كمرجي	١٩٩٧
٩- مرجع تعليمي شامل		
في صيانة الحواسب الشخصية	م. عبد الله أحمد	١٩٩٨
١٠- مرجع تعليمي في برنامج الرسم		
والتصميم الهندسي AUTOCAD 14	م. احسان مردود	١٩٩٨
١١- المرجع التدريبي الشامل لـ		
WINDOWS 98	م. إياد زوكار	١٩٩٨
١٢- ادخل إلى عالم WINDOWS 98	م. مهيب فواز النقري	١٩٩٨
١٣- الإنترنيت وإنترانيت وتصميم المواقع	م. عبد الله أحمد	١٩٩٨
١٤- تكنولوجيا المعلومات	هاني شحادة الخوري	
على أعتاب القرن الحادي والعشرين		١٩٩٨
١٥-الإدارة الاستراتيجية للشركات والمؤسسات	د.يونس حيدر	١٩٩٩
١٦-إدارة الجودة		
وعناصر نظام الجودة ISO 9004-1	م.محمد حسن -م.بسام عزام	١٩٩٩

١٧-القائد المفكر حافظ الأسد

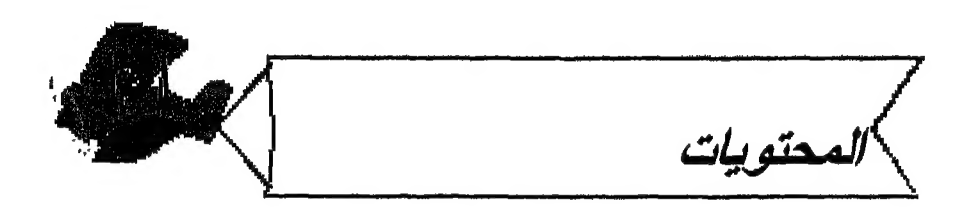
والمشروع التنموي الحضاري	د.رياض عواد-أ.هاني الخوري	١٩٩٩
١٨- فن إدارة البشر	د. محمد مرعي مرعي	١٩٩٩
١٩- المرجع الشامل لتعليمات		
برنامج AUTOCAD	م. احسان المردود -م. وهبي معاد	١٩٩٩
٢٠- الدعاية والتسويق ومعاملة الزبائن	م. حنا بللوز	١٩٩٩
٢١- المعلومياء (المعلوماتية)		
ظروفها وآثارها الاقتصادية - الاجتماعية	د. معن النقري	١٩٩٩
٢٢- المرجع الشامل لبرنامج 3D MAX	م. جورج عطا الله بركات	١٩٩٩
٢٣- دليل الجودة في المؤسسات والشركات	م. ماهر العجي - د. طلال عبود	١٩٩٩
٢٤- المرجع المفيد في علم شبكات الحواسيب	د.معتصم شفا عمري	١٩٩٩

عناوين ستصدر قريبا

اسم الكتاب	المؤلف	تاريخ النشر المتوقع
١- دراسات وبحوث التسويق	د. طلال عبود- د.حسين علي	١٩٩٩
٢- تصميم مواقع انترنيت	م. عبدالله أحمد	١٩٩٩
٣-العمل السكرتاري وبرنامج OUTLOOK	أ. بيداء الزير	١٩٩٩
٤-إدارة قواعد البيانات M.S. ACCESS	بيداء الزير-زياد كمرجي	١٩٩٩
٥- نظام الشبكات WINDOWS NT	م.عبدالله أحمد	١٩٩٩
٦- أساسيات الحوسبة والمعلوماتية	م.عبدالله أحمد	١٩٩٩
٧- أساسيات إدارة الموارد البشرية	د.محمد مرعي مرعي	١٩٩٩
٨- الدعاية والتسويق ومعاملة الزبائن- ج٢	م.حنا بللوز	١٩٩٩
٩- الكذبات العشر للعولمة	د. عدنان سليمان	١٩٩٩



Converted by Tiff Combine - (no stamps are applied by registered version)



الباب الرابع: أدوات إدارة أوراكل.

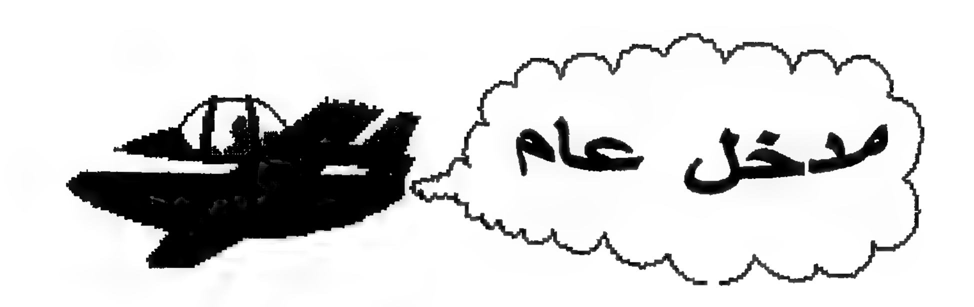
الباب الخامس: إدارة هيئة أوراكل.

الباب السادس: إنشاء قاعدة معطيات.

الباب الثامن : إدارة عناصر المخطط.

الباب الرابع عشر : الملحقات


الباب الأول



١ . مقدمة.

٢ . لمحة تاريخية عن نظام أوراكل.

٣ . ما الجديد في نسخة أوراكل٨؟

٤ . منتجات أوراكل.

٥ . مهام ومسؤوليات مدير قاعدة المعطيات.

٦ . تركيب برمجيّات أوراكل.

1



مقدّمة Introduction

نحن الآن فـــي عصر سمي بعصر المعلومات، فلقد أصبحت المعلومات هي العصب المحرك في حياتنا اليوم، خاصة بعد أن أصبح العالم قرية صغيرة بظهور ثورة الاتصالات "الإنترنت".

وفي عالم التكنولوجيا المتقدمة، فإن الحواسيب هي أفضل من يقوم بإدارة المعلومات كونها تجعل من السهل جداً تنظيم هذه المعلومات وتخزينها وحماية الهام منها.

ويمكــن للحواســيب القيــام بــإدارة المعلومات بأفضل شكل ممكن وذلك باستخدام قواعد المعطيــات Data Base، ذلك لأنها تساعد في تنظيم وتخزين المعلومات بشكل آمن، ومن ثم الحصول على هذه المعلومات بأسرع مايمكن.

وتعتــبر قواعــد المعطيــات العلاقاتية Relational Databases من أفضل أنواع قواعد المعطيــات لسهولة فهمها وتصميمها وبنائها، حيث يتم تخزين وإظهار المعلومات في هذا الــنوع من القواعد ضمن جداول Tables، إضافة إلى ذلك فإنّ قواعد المعطيات العلاقاتية

تلغي تعقيد الوصول إلى المعطيات من قبل المستخدم، مما يجعل عملية تطوير التطبيقات بسيطة نسبياً مقارنة مع أنماط أنظمة قواعد معطيات أخرى.

ثم ظهر بعد ذلك مايسمى بقواعد المعطيات غرضية التوجه Object Oriented Databases والتي تدعم التطوير غرضي التوجه مما يرفع مستوى التجريد ويسهل عملية تصميم وبناء نظام إدارة المعلومات.

أما نظم إدارة قواعد المعطيات (Data Base Management System (DBMS فهي عبارة عن برمجيات تساعد في إدارة عمليات الوصول إلى قواعد المعطيات. ويمكن لأنظمة إدارة قواعد المعطيات متعددة المستخدمين القيام بالعديد من المهام أهمها:

- إدارة الوصول المشترك إلى قاعدة معطيات وحيدة بين عدة مستخدمين، فمثلاً تقوم هذه الأنظمة بقفل lock قواعد المعطيات عندما يقوم المستخدمين بإضافة معلومات جديدة، أو تعديل المعلومات ضمنها، مما يمنع حصول تضارب بين عمل هؤلاء المستخدمين والذي يمكن أن يؤدي إلى تخريب القاعدة.
- استخدام مصادر الحواسيب بأفضل شكل ممكن مما يسمح لعدد كبير من مستخدمي التطبيقات بإنجاز أعمالهم في وقت أسرع وبأقصى إنتاجية.
- حماية معلومات قواعد المعطيات، بحيث يمكننا إعادة استرجاع العمل الذي يمكن أن يضيع لأسباب عديدة كالانقطاع المفاجئ للكهرباء أو حدوث عطل ما.

أهداف قواعد المعطيات

يمكن تحديد أهم أهداف قواعد المعطيات كما يلي:

١ـ مركزية المعلومات: في النظم التقليدية السابقة، كانت تصمم ملفات من أجل تحقيق احتياجات تطبيق معين. وبالتالي يتم تكرار المعلومات أحياناً في عدة ملفات إذا كانت تستخدمها في عدة تطبيقات وهذا يؤدي إلى ضياع أمكنة بسبب تكرار المعلومات، واحتمال التضارب في المعلومات، كذلك صعوبة قراءة المعلومات نظراً لوجود عدة ترميزات لنفس المعلومة.

لذلك كان الهدف الأهم لقواعد المعطيات هو مركزية المعلومات من أجل إلغاء التكرار وإدخال المعلومات مرة واحدة فقط ولتخفيف مركزية المراقبة والتحكم.

٢ـ **استقلالية المعلومات عن المعالجة**: يجب أن تكون هناك استقلالية في المعطيات عن البرامج التي تستخدم هذه المعطيات لأن أي تغيير في ملف معطيات بعض البرامج سيؤدي إلى تغيير البرامج المشتركة في هذا الملف، فمثلاً إذا كان لدينا ملف ذاتية للموظفين وكان لدينا برنامج أول لحساب الرواتب وثان للضمان الصحي، فإن أياً من هذين البرنامجين يجب ألا يؤثر على المعطيات الموجودة في ملف ذاتية الموظفين.

بالتالي يجب أن تكون هناك قاعدة معطيات تحتوي كل المعطيات المتعلقة بالمؤسسة ويجب أن تكون ملائمة ومواكبة لتطور المؤسسة مع إمكانية إيجاد تطبيقات مختلفة تتعامل مع نفس المعطيات، ويجب ألا يتأثر أي تطبيق بالتعديل الذي يجريه تطبيق لآخر على طريقة رؤيته للمعلومات.

٣ـ **تحديد الارتباطات بين المعطيات**: فمثلاً في مستودع ما إذا كان لدينا معلومات عن محتويات المستودع ومعلومات عن الموزعين، فيجب ربط كل موزع مع قائمة المنتجات التي يوزعها، وبالعكس يجب ربط أي منتج مع قائمة الموزعين، مع تحديد شروط البيع لكل موزع أو منتج.

٤ـ **تكامل المعطيات**: بعد التأكد من مركزية المعلومات وصحة عمل المؤسسة، يجب التأكد من تكامل وانسجام المعلومات وتحديد الشروط التي تضمن صحة المعلومات وعدم تضاربها، من أجل ذلك يجب تحديد المعايير التي تضمن التكامل في المعطيات. ويوجد نوعان من المعايير:

ـ **معايير ثابتة**: فمثلاً يجب أن لا يتجاوز عمر الموظف ١٥٠ سنة.

ـ **معايير متغيرة**: مثلاً الراتب بعد الترفيع أكبر من الراتب قبل الترفيع.

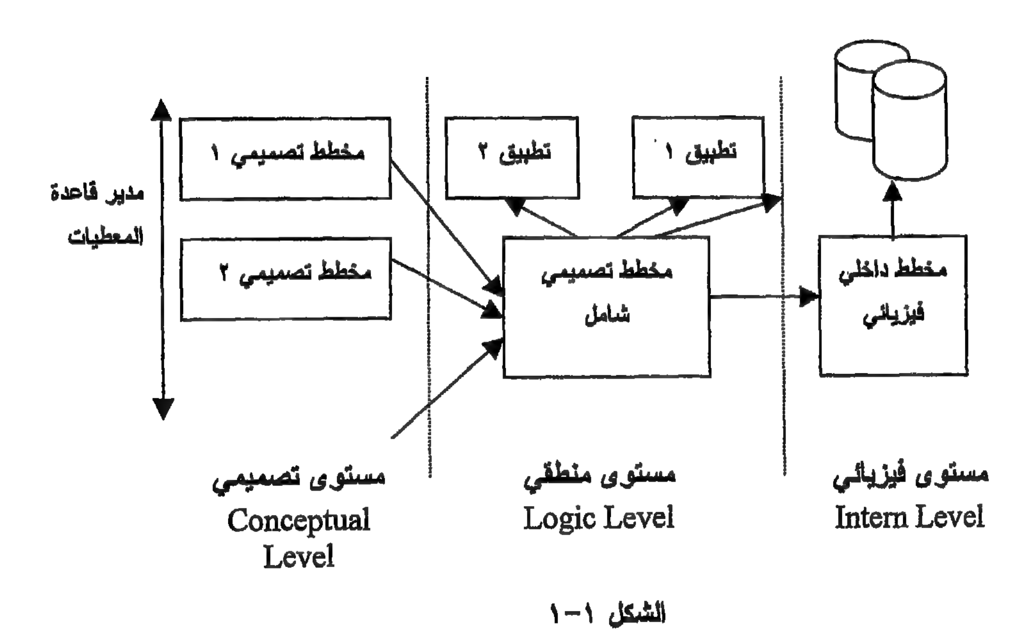
٥ـ **أمان المعطيات**: وهي مجموعة الأدوات التي تسمح بضمان صحة المعلومات وعدم ضياعها في حال حدوث خطأ أو عطل ما.

٦ـ **الوثوقية**: تسمح بتنظيم سماحية الوصول إلى المعطيات من قبل عدة مستخدمين، كتحديد كلمة سر أو عرض أو تحديد العملية التي يمكن أن يقوم بها المستخدم وحسب أهميتها (إمكانية الإنشاء، الاستفسار، التعديل ..... إلخ).

٧ـ **إدارة الاستعمال المشترك للمعطيات**: من أجل إلغاء عدم حدوث تضارب في المعلومات عند قيام أكثر من مستخدم بالتعامل مع نفس الملف خاصة عند إجراء عمليات التعديل أو الحذف أو الإنشاء.

البنية العامة لقواعد المعطيات

يمكن توضيح البنية العامة لقواعد المعطيات على الشكل ١-١:



الشكل ١-١

حيث يتم في المستوى التصميمي دراسة المؤسسة من أجل تحديد مهام جزئية مستقلة نوعاً ما، وتحديد المعطيات والارتباطات بين هذه المعطيات لكل من المهام.

أما في المستوى المنطقي فيتم دمج المخططات التصميمية المختلفة واستنتاج مخططاً منطقياً وحيداً، مع مراعاة عدم تكرار المعطيات وإيجاد الارتباطات الشاملة بين هذه المعطيات، ويتم التعبير عن هذا المخطط بواسطة لغة تسمى **لغة توصيف المعطيات DDL** (Data Definition Language).

وبعد إيجاد المخطط التصميمي الشامل يتم إنشاء التطبيقات المختلفة التي يحقق تنفيذها المباشر ماهو مطلوب من قاعدة المعطيات. وتتم كتابة هذه التطبيقات بواسطة لغة تسمى **لغة التعامل مع المعطيات DML** (Data Manipulation Language)

وأخيراً في المستوى الفيزيائي فيتم تخزين المعطيات بشكلها الفعلي ضمن الحاسوب وبترميز معين خاص بقاعدة المعطيات.

وظائف نظم إدارة قواعد المعطيات

يجب أن يتوفر في أي نظام إدارة قواعد معطيات مجموعة من الوظائف أهمها :

١- إدارة حجم كبير جداً من المعلومات وبشكل موثوق.
٢- إمكانية العمل في بيئة متعددة المستخدمين.
٣- حماية البيانات من الذين لا يملكون حق الدخول إلى قاعدة المعطيات.
٤- تقديم الأدوات لإصلاح الأعطال والقيام بالتخزين الاحتياطي.

أهم نظم إدارة قواعد المعطيات

يمكن تحديد أهم نظم إدارة قواعد المعطيات على الشكل التالي:

١ــ نظم إدارة قواعد المعطيات الهرمية: IMS،SYSTEM 2000 .
٢ــ نظم إدارة قواعد المعطيات الشبكية: SOCRATE , TOTAL.
٣ــ نظم إدارة قواعد المعطيات العلاقاتية: INGRES,ORACLE.
٤ــ نظم إدارة قواعد المعطيات غرضية التوجه: POSTGRESS, O_2.

نظام إدارة قواعد المعطيات Oracle

يعتبر نظام أوراكل ORACLE أحد أفضل أنظمة إدارة قواعد المعطيات عالية المستوى، وهو يتميز بما يلي:

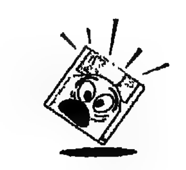
- إمكانية إدارة قواعد معطيات كبيرة جداً Very Large Data Bases (VLDB) يمكن أن تصل حتى تيرابايت من المعلومات.
- إمكانية التعامل مع عدد كبير من المستخدمين بشكل متزامن يمكن أن يصل حتى عشرات الآلاف من المستخدمين.

- نظام موثوق فيه : في بعض المواقع يعمل نظام أوراكل على مدار اليوم كله وبدون توقف، بالإضافة إلى كونه قادراً على تجاوز الأعطال جميعها حتى الفيزيائية.
- يقدم حماية للمعلومات من الدخول غير الشرعي، فهو يحتوي على الكثير من الأدوات التي تسمح بتحديد و مراقبة الدخول إلى قواعد المعطيات وبمستويات مختلفة ومتنوعة.
- الأداء العالي : إن كل الميزات السابقة لا تبطئ من عمل النظام والأداء يبقى مرتفعاً.
- مواصفات قياسية : يلتزم أوراكل بالمقاييس الموضوعة للغات الوصول للبيانات، وواجهات التخاطب مع المستثمر.
- إمكانية النقل : فهو مصمم ليعمل على أنظمة تشغيل مختلفة وحواسيب مختلفة، وبنيته متشابهة في جميع الأنظمة، وأي تطبيق يمكن نقله إلى أي حاسوب آخر.
- الاتصالات : فهو يسمح لأنواع مختلفة من الحواسيب وأنظمة التشغيل بالمشاركة في البيانات إذا كانت موصولة على شبكة.

لذلك يعتبر نظام أوراكل أحد أفضل أنظمة إدارة قواعد المعطيات والتي تصنّف ضمن أنظمة الزبون/المخدّم Client/Server. وهو يعمل ضمن العديد من نظم التشغيل مثل UNIX وWindows95 وWindows NT و Novel. كما يتميز بإمكانية استخدام تطبيقات الوسائط المتعددة كالتطبيقات التي تتعامل مع الصور والأصوات والمخططات البيانية وغيرها.

وهو يحتوي على وسائل تطوير باللغة العربية تتوافق مع تعريب ويندوز بشكل مباشر. كما يمتلك العديد من أدوات تطوير التطبيقات منها CDE II و Developer 2000 و Designer 2000 والتي تسمح بإنشاء نماذج وتقارير بشكل ممتاز وبأكثر من لغة في الوقت نفسه.

الرموز المستخدمة في الكتاب

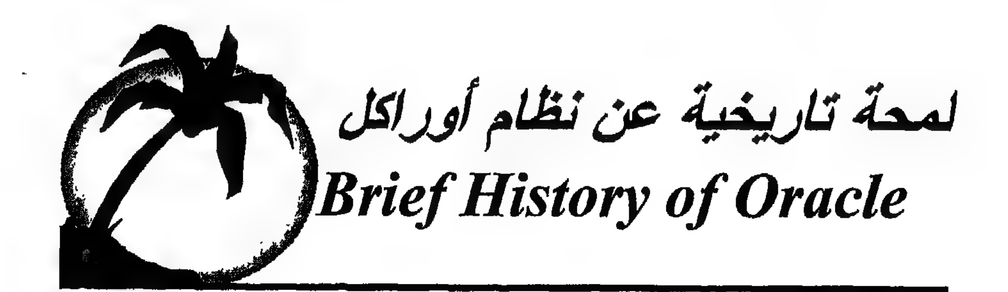


ملاحظة



مثال





لمحة تاريخية عن نظام أوراكل
Brief History of Oracle

في عام ١٩٧٧ قامت شركة اسمها Relational Software (RSI) Incorporated ببناء نظام إدارة قواعد معطيات بالاسم Oracle، حيث باعت النسخة الأولى من هذا النظام في عام ١٩٧٩.

أما النسخة الثانية منه فقد تمّ بناؤها من أجل العمل على حواسيب Digital PDP-11 والتي تعمل على نظام التشغيل RSX-11 والتي عملت فيما بعد على نظام DEC VAX. النسخة الثالثة من النظام، تمّ طرحها عام ١٩٨٣ حيث أجريت عليها الكثير من التحسينات، خاصّة تلك المتعلقة بلغة SQL. ولقد كتبت هذه النسخة كاملة بلغة C وتمّ تغيير اسم الشركة من RSI إلى Oracle Corporation.

النسخة الرابعة من أوراكل تمّ إنجازها عام ١٩٨٤، ولقد دعمت هذه النسخة نظامي التشغيل VAX و IBM VM كما كانت أول نسخة تدعم خاصية تناسق القراءة Read Consistency.

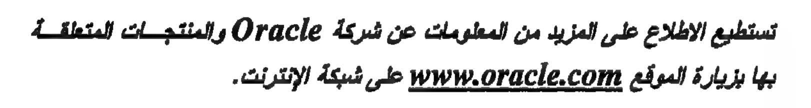
النسخة الخامسة من أوراكل أنجزت عام ١٩٨٥، ولقد قدمت هذه النسخة دعماً كبيراً لهذا النظام لكونها أضافت تقنية الزبون/المخدّم Client/Server باستخدام SQL*NET، كما أنها أول منتج يعمل ضمن نظام MS-DOS استطاع كسر حاجز 640 KB.

النسخة السادسة من أوراكل أنجزت عام ١٩٨٨، ولقد أضافت تقنية القفل من مستوى منخفض low-level locking، بالإضافة إلى العديد من التحسينات والوظائف الجديدة مثل توليد السلاسل. ولقد عملت هذه النسخة على الكثير من الأنظمة والمنصات Platforms، كما أضيف إليها خيار التوازي Parallel Option وذلك على نسخة Oracle6.1 والتي تعمل على نظام DEC VAX وذلك عام ١٩٩١، ومن ثمّ أصبح هذا الخيار متاحاً ضمن العديد من المنصّات.

أما نسخة Oracle7 فلقد طرحت عام ١٩٩٢، وتمّ فيها إجراء العديد من التغييرات خاصة المتعلقة بمنطقة الذاكرة والمعالج واستخدام الدخل والخرج. إضافة إلى ذلك فقد احتوت هذه النسخة الكثير من الأدوات المساعدة مثل SQL*DBA.

أخيراً، تمّ طرح النسخة الأخيرة من أوراكل Oracle8 عام ١٩٩٧، حيث أضيف إليها مفهوم العناصر Objects، بالإضافة إلى العديد من التقنيات وأدوات إدارة القاعدة.





*تستطيع الاطلاع على المزيد من المعلومات عن شركة **Oracle** والمنتجات المتعلقة بها بزيارة الموقع **www.oracle.com** على شبكة الإنترنت.*





ماالجديد في نسخة أوراكل٨؟
Oracle8 New Features

تمّ بناء النسخة Oracle8 بالاعتماد على قوة النسخة السابقة Oracle7 بشكل كبير والتي صدرت عام ١٩٩٣ وكانت أحد أفضل أنظمة إدارة قواعد المعطيات العلاقاتية. أما Oracle8 فهو عبارة عن نظام إدارة قواعد معطيات علاقاتية غرضية التوجه Object-Relational Data Base (ORDBMS) Management System.

فهو إذاً عبارة عن مخدّم قواعد معطيات يمتلك إمكانيات نظام إدارة قواعد معطيات علاقاتية بالإضافة إلى إمكانيات نظام قواعد معطيات غرضية التوجه.

سنعرض فيما يلي أهم الأمور والإمكانيات التي أتى بها Oracle8:

الجداول والفهارس المجـــزأة Partitioned Tables and Indexes

تساعد هذه الإمكانية الجديدة في تقليل المشاكل التي يمكن أن تحدث بشـــكل خـــاص مـــع الجداول والفهارس الكبيرة جداً، لنفترض مثلاً السيناريو التالي الذي يمكن أن يحدث بسـبب كبر حجم المعطيات :

- استعلام معين يتطلب مسحاً كاملاً للجدول هائل الحجم، وهذا يزيد كثيراً من عنـــاء النظام والتطبيق عند انتظار أوراكل ليتم قراءة العدد الهائل من كتل معطيات الجدول.
- حالة استثنائية تتطلب التعامل مع جدول كبير واحد. وبسبب عطل ما في القـوص، يصبح هذا الجدول غير متاح بسبب عدم إمكانية الوصول إلى كتلة معطيـــات واحـــدة فقط. الحل الوحيد هنا هو أن يقوم مدير النظام بعملية استرداد لكامل الفضاء الجدولـــي الذي يحتوي هذا الجدول قبل أن يصبح بالإمكان التعامل معه من جديد.

لذلك فإنّ عملية تجزيء المعطيات تسمح لأوراكل بتخزيـــن الجـــداول الكبـــيرة، إضافـــةً للفهارس المرتبطة بها، وذلك ضمن أجزاء صغيرة سهلة المعالجة بالنسبة لكتلة معطيـــــات كبيرة وذلك للأسباب التالية :

- تحسّن أداء التطبيقات، وذلك لأن أوراكل يمكنه الوصول إلى عدة أجزاء معطيـــات بشكل متوازٍ.
- إمكانية الوصول إلى أجزاء الجداول السليمة حتى لو تعطل القرص وأدى إلى عدم إمكانية الوصول إلى جزء أو أكثر منه.
- نظراً لأن أجزاء الجداول عبارة عن مناطق تخزين منفردة، يمكن لمديـــر قـــاعدة المعطيات إدارة الأجزاء بشكل منفصل خاصة لإجراء بعض العمليات الخاصة كالنسخ الاحتياطي لقاعدة المعطيات أو استردادها.

إدارة عدد كبير من المستخدمين Management of Large User Population

كما نعلم فإن العدد الكبير من المستخدمين المتزامنين يؤثر بشكل سلبي في أداء النظام، من أجل ذلك يحتوي أوراكل ضمن بنية الإجراءات على تقنية المخدّم متعدّد التشعّبات (MTS) MultiThread Server والتي تساعد بشكل كبير في حلّ هذه المشكلة، وتمكنه من إدارة المئات وحتى الآلاف من الزبائن بشكل فعّال. إضافة إلى ذلك تمتلك نسخة Oracle8 العديد من التقنيات الجديدة وخاصة المتعلقة ببيئة الزبون/المخدّم كبرمجيات Net8 التي تسمح بإدارة مصادر الشبكة بشكل فعّال جداً.

التنفيذ اللاحق للطلبات Deferred Requests Execution

في أغلب تطبيقات قواعد المعطيات التقليدية ضمن بيئة الزبون/المخدّم Client/Server، تحاول هذه التطبيقات تنفيذ طلبات مستخدمي قاعدة المعطيات مباشرة وفوراً بعد الضغط على زر OK. ينتظر بعدها المستخدمون نتائج طلباتهم قبل الاستمرار بعملهم.

لكن تحتاج بعض الأنظمة إلى قيام التطبيقات بالتنفيذ اللاحق deferred execution لطلبات المستخدم وذلك باستخدام الأرتال Queues، مما يسمح لها بإنجاز الأعمال الصعبة والمعقدة بشكل أمثل خلال ساعات العمل العادية.

من أجل ذلك أوجد Oracle8 تقنية جديدة وهي تقنية الأرتال المتقدمة Advanced Queuing وذلك من أجل التطبيقات التي تحتاج إلى تأخير تنفيذ التحويلات لقاعدة المعطيات والتي يتم حمايتها تلقائياً باستخدام أدوات النسخ الاحتياطي والاسترداد.

المعالجة المتوازية Parallel Processing

لقد أصبح العديد من التطبيقات بحاجة إلى استخدام مايسمى بالمعالجـــة المتوازيـــة لدعـــم عملها، يوفر Oracle8 ذلك من خلال تقنيـــة الاســـتعلام المتـــوازي Parallel Query، والمخدّم المتوازي Parallel Server.

الاستعلام المتوازي Parallel Query

تسمح هذه التقنية لأوراكل باستخدام جميع الإمكانيات المتاحة من قبل إجـــراءات المخـــدّم، والتي تعطي زمن استجابة ممتاز حتى في الاستعلامات المعقدة جداً والتي تشغل المعـــالج CPU والدخل/الخرج بشكل كبير.

يقوم أوراكل بتقسيم استعلام SQL إلى استعلامات جزئية، ثم يسمح للإجراءات المتعـــددة بمعالجة هذه الاستعلامات الجزئية بشكل متوازٍ. يقوم بعدها بدمـــج نتـــائج الاســـتعلامات الجزئية وإعطاء النتيجة النهائية للمستخدم بشكل سريع.

يقوم Oracle8 أيضاً بدعم تنفيذ جميع عمليات لغة معالجـــة المعطيـــات Data (DML) Manipulation Language وبشـــكل متـــوازٍ، وتتضمـــن عمليـــات INSERT و UPDATE و DELETE. كذلك يمكن لجميع الاستعلامات، بما فيها تلك التـــي تحتـــاج لمسح الفهارس أن تعمل بشكلٍ متوازٍ، ويقوم Oracle8 بالاستفادة من مـــيزات الجـــداول والفهارس المجزأة في إجراء عمليات الاستعلام المتوازي.

المخدّم المتوازي Parallel Server

يسمح هذا الخيار لعدة مخدّمات قواعد معطيات أو عدة هيئات بالعمل على عقـــد مختلفـــة، وذلك من أجل الوصـــول بشكلٍ متزامن إلى قاعدة معطيات معينة.

ويحتوي Oracle8 العديد من التقنيات التي تسمح بتحسين أداء النظام كتقنية مديـــر القفـــل الموزّع Distributed Lock Manager، والفهارس ذات المفتاح المعـــاكس Reverse-Key Indexes.

أوراكل٨ وقواعد المعطيات غرضية التوجه

يعتبر Oracle8 من أفضل أنظمة إدارة قواعد المعطيات العلاقاتية غرضية التوجه، وهو يتضمن العديد من التقنيات والأنماط الجديدة العلاقاتية-الغرضية كأنماط العناصر Object Types، والطرق Methods، وعناصر المشاهد Object Views.

كما أضاف Oracle8 أنماط معطيات جديدة من أجل التطبيقات متعددة الوسائط مثل Large Object Binary (LOB) والذي يمكن تخزينه في مقطع مختلف عن مقطع معطيات الجدول ضمن نفس قاعدة المعطيات، أو خارج قاعدة المعطيات لتجنب توليد مقاطع تراجع وملفات إرجاع خاصة به.

تسمى بيئة التطوير غرضية التوجه ضمن Oracle8 بسيدونا Sedona.

النسخ الاحتياطي والاسترداد Oracle8 Backup and recovery

أتى Oracle8 بالعديد من التقنيات الجديدة المتعلقة بعمليات النسخ الاحتياطي والاسترداد، وذلك لجعل هذه العمليات أسهل وأسرع، فهو يقوم بتخزين معلومات تفصيلية حول النسخ الاحتياطي الأخير الذي تمّ إجراؤه على قاعدة المعطيات، بحيث يستطيع تحليل حالة قاعدة المعطيات بشكل تلقائي وإجراء الاسترداد التلقائي للقاعدة المعطوبة، مما يقلل من احتمال حصول الخطأ البشري عند إجراء هذه العمليات.

كما يدعم Oracle8 عمليات النسخ الاحتياطي المتزايدة Incremental Backup، مما يسمح بتخفيف زمن النسخ الاحتياطي، وتقليص الحجم اللازم، لأنه يقوم بإجراء النسخ الاحتياطي لكتل المعطيات التي تمّ تغييرها منذ آخر عملية نسخ احتياطي فقط.

إضافة إلى ذلك أصبح بإمكان Oracle8 استرداد كامل قاعدة المعطيات، بالإضافة إلى إمكانية استرداد فضاءات جدولية منفردة بنمط Point-In-Time.

أخيراً فلقد أضاف Oracle8 العديد من التقنيات الجديدة المتعلقة بقواعد المعطيات الموزّعة والتي تقلل من كمية المعطيات التي يتمّ نشرها على الشبكة.

١





منتجات أوراكل
Oracle Products

سنقوم في هذا الفصل وكجزء من التعريف بنظام أوراكل، بإلقاء نظـــرة ســـريعة على الخيارات المتاحة لاستخدام منتجات أوراكل. وعلى الرغم مـــن أننـــا سنقوم بشرح مفصل للعديد من هذه المنتجات في فصولنا القادمة، إلا أنـــه من الأفضل التعريف بها لكي تتكون لدينا فكرة كاملة عن الإمكانيات والأدوات والتقنيـــات التي نستطيع الاستعانة بها مع نظام أوراكل.

تنقسم هذه المنتجات إلى ثلاثة مجالات:

- مخدّم أوراكل Oracle Server.
- أدوات التطوير Development tools.
- التطبيقات Applications.

مخدّم أوراكل Oracle Server

وهو عبارة عن نظام إدارة قواعد المعطيات نفسه، ويتضمن العديد من الخيارات والميزات، كخيار الاستعلام المتوازي Parallel Query وبرتوكولات الشبكة Networks Protocols وخيارات إدارة النظام المتقدمة Advanced System Administration

من الخيارات الأساسية المتاحة مع مخدّم أوراكل:

١- **Enterprise Manager**: وهو أحد الخيارات المهمة الجديدة مع أوراكل ويتكون من الوكلاء الأذكياء intelligent agents وإدارة الكونسول management console .

إدارة الكونسول هو الجزء الرئيسي في حزمة الإدارة الرسومية الجديدة في أوراكل، وهو يعمل فقط ضمن نظام WINDOWS NT، لكن باستطاعته إدارة أي مخدّم أوراكل، كما يسمح لمدير قاعدة المعطيات DBA بالتحكم وبشكل رسومي بنظام أو أكثر من أنظمة أوراكل.

كما أنه يستخدم لتوصيف و إدارة هيئات أوراكل، وتشخيص الأخطاء ويمكن توصيفه لتحذير مدير قاعدة المعطيات في حال حدوث مشكلة.

أما الوكلاء الأذكياء فهم أحد أهم مفاتيح Enterprise Manager، وهم يعملون ضمن مخدّم أوراكل ويقدمون طبقة الاتصال الضرورية للكونسول من أجل الاتصال مع هذه الأنظمة. كما ستخدمون بروتوكولات Simple Network Management) SNMP Protocols) للاتصال مع هذا الكونسول.

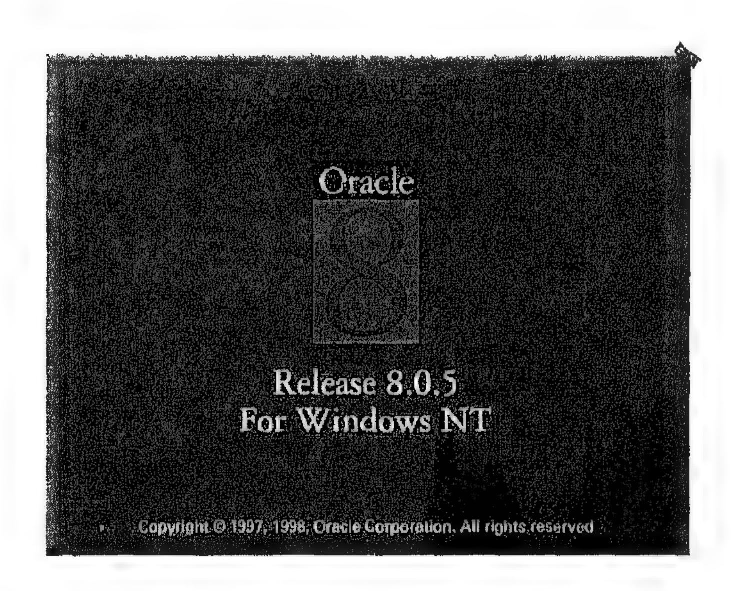
٢- **ConText**: يمكن استخدامه مع الأنظمة النصية text systems. وهو يساعد في تحليل وترشيح النصوص لتسريع عملية القراءة وإظهار الملخصات. كما يمكن استخدامه لإجراء تحليل مفصل عن النص الذي يقوم بمعالجته والتحقق من الأخطاء القواعدية كذلك نوعية ونمط الكتابة.

٣- **Media Server**: وهي تتضمن مكتبة متعددة الوسائط Multimedia library عالية المستوى، تساعد في إجراء مختلف عمليات تخزين واستخراج وإدارة الأفلام والمقاطع الموسيقية و الصور.

٤- **Spatial Data**: يستخدم هذا الخيار لإدارة قواعد المعطيات التي تحتوي على معلومات فضائية. وهو يسمح بتخزين معطيات جغرافية ومعطيات فضائية أيضا.

٥- **Oracle Web Server**: وهو يسمح بتوفير خدمة الوصول إلى قاعدة معطيات أوراكل من قبل شبكة وب العالمية World Wide Web بدلا من الحصول على هذه المعطيات من الملفات التقليدية.

٦- **Internet Commerce Server**: وهو عبارة عن مجموعة كاملة من الأدوات المصممة لمساعدتك على بناء وتنفيذ وإدارة نظم أوراكل المستخدمة للتجارة ضمن شبكة وب بطريقة ممتازة و آمنة.



أدوات التطوير Development Tools

تعتبر أدوات التطوير أحد أقوى ما تتميز بها شركة أوراكل نظراً لمرونتها وميزاتها المتكاملة. فعندما ظهرت أنظمة المخدّم/الزبون Client/Server في بداية ١٩٩٠، سرعان ما توافقت معها أدوات أوراكل. وعندما ظهرت تطبيقات جافا وHTML في منتصف ١٩٩٠، سرعان ما توافقت معها أدوات أوراكل مرة أخرى.

يوجد العديد من أدوات التطوير ضمن نظام أوراكل أهمها:

١- **Oracle Designer**: وهي عبارة عن مجموعة من أدوات النمذجة Modeling tools والتي تخفّف وإلى حدّ كبير من عناء تصميم الأنظمة. يمكن أن تستخدم هذه الأدوات لتوفير دخل لأدوات Oracle Developer وتطوير النماذج الشائعة المستخدمة في إجراءات الأعمال.

٢- **Oracle Developer**: وهي مجموعة من الأدوات التي تسمح لك ببناء التطبيقات للعمل مع أنظمة Motif, Macintosh, Windows وأنظمة نمط المحارف Character mode.

وهي تقوم بتضمين المخططات والصور كما أنها تدعم العناصر متعددة الوسائط كالأفلام والموسيقى وغيرها.

٣- **Oracle Discoverer**: وهي أداة لتحليل المعطيات data analysis التي تدعم الاستعلامات والتقارير والتحليل البياني متعدد الأبعاد للمعطيات.

٤- **Extra Objects**: عبارة عن أداة لتطوير واجهة المستخدم الرسومية GUI. وهي متاحة ضمن أنظمة Windows و Macintosh و OS/٢ والتي تسمح بتطوير التطبيقات التي تستخدم نسبة قليلة من مصادر النظام.

وهي أداة مشابهة لـ Developer ٢٠٠٠ لكنها تفتقر للعديد من الميزات التي يمتلكها.

٥- **Objects for OLE**: وهي مجموعة من الأدوات تسمح لك بربط تطبيقات OLE_ compliant مع نظام إدارة قواعد المعطيات العلاقاتية أوراكل، فهي تقدم طريقة سريعة وسهلة لاستكشاف قوة التطبيقات كصفحات العمل Spread sheets. كما أن عناصر OLE تسمح بربط جداول قاعدة المعطيات مع مستندات معالج النصوص بطريقة سهلة.

٦- **Oracle Programmer**: مجموعة من الأدوات التي تساعد في تطوير SQL و PL/ SQL والإجرائيات المخزنة وهي مساعدة لمطوري التطبيقات.

٧- **Media Objects**: أداة مساعدة في تطوير التطبيقات متعددة الوسائط وهـي تدعـم بيئة المخدّم/الزبـون Client/Server والأقـراص المدمجـة CD_ROM وإجـراءات التلفزيون التفاعلية Interactive Television Processes.

٨- **Database Designer**: نسخة من منتج Oracle Designer للمساعدة في تصميـم وبناء قواعد المعطيات. وهي أداة مستخدم وحيد Single User Tool تقوم بشكل رسومي بتصميم جداول قاعدة المعطيات وتوليد تعليمات SQL التي يمكن استخدامها لإنشاء هـذه القاعدة.



التطبيقات Applications

هناك نمطان من أنماط تطبيقات أوراكل:
التطبيقات التقليدية traditional applications، وتطبيقات (Online OLAP Analytical processing).

التطبيقات التقليدية *Traditional Applications*

وهي مجموعة التطبيقات المستخدمة لإجراء المهام الأساسية للأعمال والمستخدمة في العديد من كبريات الشركات العالمية. تدعم هذه التطبيقات العديد من المجالات المالية وإدارة المشاريع والمبيعات والتصنيع وغيرها.

تطبيقات *OLAP*

وهي عبارة عن تطبيقات تزودنا بواجهة رسومية لتطبيقات data-ware housing و DSS. كما أنها تعطينا نموذجاً متعدد الأبعاد لقاعدة المعطيات يمكّننا من إجراء العمليات الإحصائية والتحليلية.







مهام ومسؤوليات مدير قاعدة المعطيات
Oracle8 DBA Responsibilities

سنقوم في هذا الفصل بشرح المهام الأساسية لمدير قاعدة معطيات أوراكل DBA وخاصة بالنسبة للتطبيقات الكبيرة، بالإضافة إلى ذلك سنقوم بتحديد أهم الأدوات الأساسية التي يمكن لمدير القاعدة استخدامها لتسهيل عمله.

المهام الأساسية لمدير قاعدة معطيات أوراكل

نظراً لأن قاعدة معطيات أوراكل يمكن أن تصل لأحجام كبيرة ويمكن للعديد من المستخدمين التعامل معها، يتحتم وجود شخص أو مجموعة أشخاص لإدارة القاعدة. والذي يسمى مدير قاعدة المعطيات DBA (DataBase Administrator).

من أهم المهام التي يقوم بها مدير قاعدة المعطيات:

١- تركيب برمجيات أوراكل Install ORACLE Software: تعتبر هذه المهمة من أولى المهام التي يفترض بمدير قاعدة المعطيات معرفتها ويجب عليه تركيب برمجيات مخدّم أوراكل إضافة إلى الأدوات الخاصة بأوراكل كذلك التطبيقات المستخدمة للوصول إلى قاعدة المعطيات.

٢- الاستفادة من تجهيزات مخدّم قاعدة المعطيات: بعد تثبيت برمجيات أوراكل يجب القيام بدراسة كيفية الاستفادة من مصادر الجهاز المتاحة وبالشكل الأمثل من أجل تطبيقات أوراكل، وهي تشمل:

- ما هي سواقات الأقراص المتاحة لأوراكل وقواعد المعطيات؟
- كم عدد الأشرطة Tape المتاحة لتشغيل أوراكل؟

٣- تخطيط قاعدة المعطيات Plan The database: كمدير لقاعدة المعطيات يجب عليك تخطيط ما يلي:

- بنى التخزين المنطقية لقاعدة المعطيات.
- تصميم قاعدة المعطيات بشكل كامل.
- استراتيجية إجراء نسخ احتياطي لقاعدة المعطيات.

٤- إنشاء وفتح قاعدة المعطيات Create and Open The Database.

٥- تنفيذ تصميم قاعدة المعطيات: بعد أن تقوم بإنشاء قاعدة المعطيات وتشغيلها، يمكنك إنشاء البنية المنطقية المخططة لقاعدة المعطيات وذلك بإنشاء جميع الفضاءات الجدولية وجميع مقاطع التراجع الضرورية، ومن ثم إنشاء جميع العناصر الضرورية لقاعدة المعطيات.

٦- **إنشاء نسخة احتياطية لقاعدة المعطيات Backup The Database**: بعد أن تقوم بتصميم بنية قاعدة المعطيات، يجب عليك تنفيذ استراتيجية النسخ الاحتياطي التي قمت بالتخطيط لها من قبل وذلك بإضافة ملفات إرجاع للقاعدة ووضع أول نسخة احتياطية لقاعدة المعطيات ضمن هذه الملفات وتوقيت عمليات النسخ الاحتياطي المستقبلية في أوقات منتظمة.

٧- **إضافة حسابات مستخدمين جدد Enroll System Users**: بعد أن تقوم بإنشاء نسخة احتياطية لبنية قاعدة المعطيات، يمكنك البدء بإضافة مستخدمين جدد للقاعدة وإعطائهم السماحيات الخاصة بهم.

٨- **تحسين أداء قاعدة المعطيات Tune Database Performance**.



أدوات مديـــر قـــــاعدة المعطيــــــات Data Base Administrator Utilities

توجد مجموعة من الأدوات المتاحة التي تساعدك في العمل والتحكم بمخدّم أوراكل أهمها:

١- **Server Manager**: تسمح لك هذه الأداة بمراقبة والتحكم بقاعدة معطيات أوراكـل. باستخدام أوامر SQL القياسية.

٢- **Enterprise Manager**: تعتـبر الأداة Oracle Enterprise Manager أحـد الأدوات الجديدة التي أصدرتها شركة أوراكل للسماح لمدير قاعدة المعطيـــات DBA بإدارة ممثلي أوراكل باستخدام واجهات رسومية، مما يسمح بإظهار المعلومات بشــكل أفضل.

٣- **SQL*LOADER**: يمكن استخدامها من قبل مدير قاعدة المعطيات كذلك من قبـل مستخدمي أوراكل. تساعدنا هذه الأداة على شحن المعطيات من ملفات نظام التشـــغيل القياسية (على شكل Text أو معطيات بتنسيق معيّن) إلى جـــداول قـــاعدة معطيـــات أوراكل.

٤- **EXPORT& IMPORT**: تستخدمان لنقل معطيات موجودة تنسيق أوراكل مـــن وإلى قواعد معطيات أوراكل.

يمكن مثلاً استخدام الملفات التي تمّ استيرادها من أجل أرشفة معطيات القـــاعدة، أو نقـل معطيات بين قواعد معطيات أوراكل متعددة تعمل ضمن نفس نظام التشغيل أو في أنظمـــة تشغيل مختلفة.





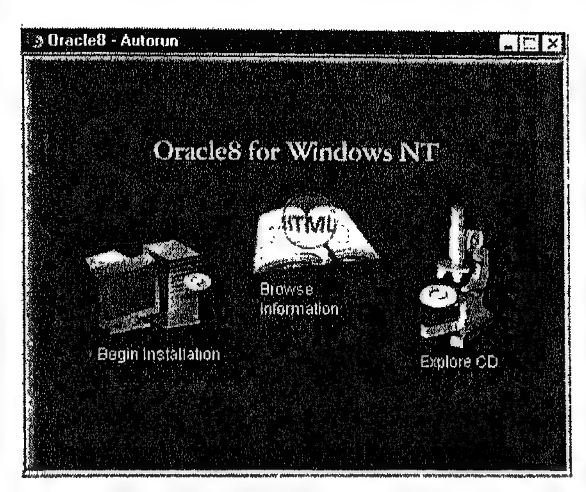
تركيب برمجيات أوراكل
Installing Oracle Software

سنقوم في هذا الفصل بتعلّم كيفية تركيب برمجيات أوراكل وذلك على شبكة Windows NT، حيث سنشرح الخطوات اللازمة لتركيب نسخة Oracle8 على الحاسب المخدّم، كذلك تركيب زبون أوراكل على الحواسيب الزبائن.

طبعاً عملية تركيب برمجيات أوراكل على نظام Windows NT لاتختلف كثيراً عن عملية التركيب على أنظمة التشغيل الأخرى.

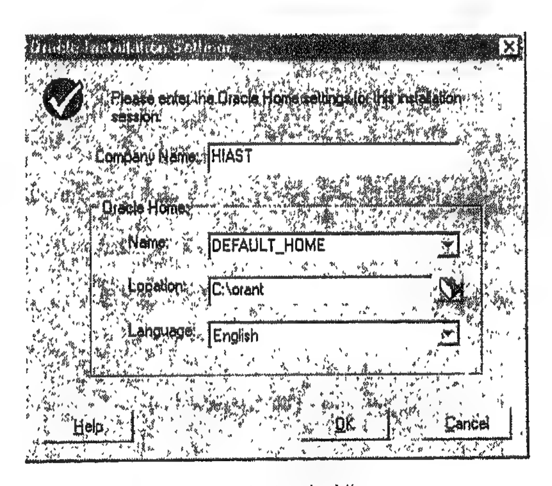
تركيب مخدّم أوراكل Installing Oracle8 Server

عندما تقوم بوضع القرص المدمج الخاص بنسخة Oracle8 في سواقة مخدّم Windows NT Server ، يتم تلقائياً تشغيل برنامج التركيب كما في الشكل ٦-١.

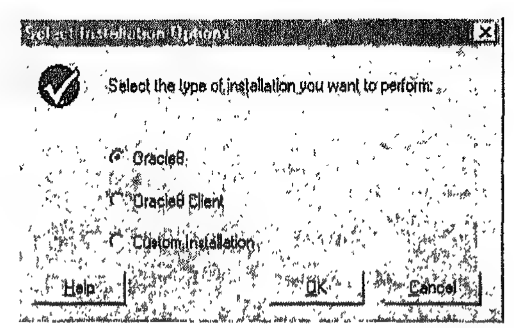


الشكل ٦-١

انقر زر Begin Installation لبدء عملية التركيب حيث تظهر نافذة تشبه الشكل ٦-٢، يتم في هذه النافذة تحديد اسم الشركة التي تمتلك النسخة Company Name، كذلك موقع أوراكل Oracle Home وذلك بتحديد اسم الموقع Name، واسم المجلّد الذي سيتم وضع برمجيات أوراكل فيه مع موقع هذا المجلد Location، أخيراً قم بتحديد لغة التطبيق Language.



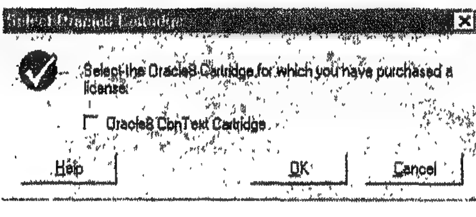
الشكل ٦-٢



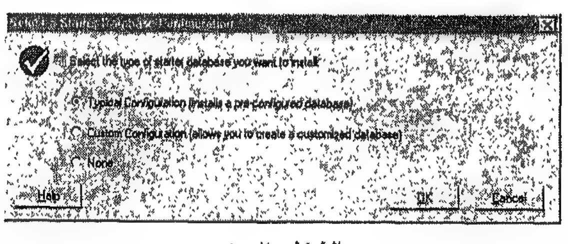
الشكل ٦-٣

انقر زر Ok يظهر صندوق حوار جديد يشبه الشكل ٦-٣ لتحديد نمط التركيب، وباعتبار أننا نقوم هنا بتركيب نسخة مخدّم أوراكل لذلك نختار الخيار الأول Oracle8.

انقر زر Ok يظهر صندوق حوار جديد يشبه الشكل٦-٤ يتم فيه اختيار خرطوشة أوراكل التي تمّ شراؤها مع نسخة أوراكل، حدد صناديق التحقق المطلوبة، ثم انقر زر Ok.



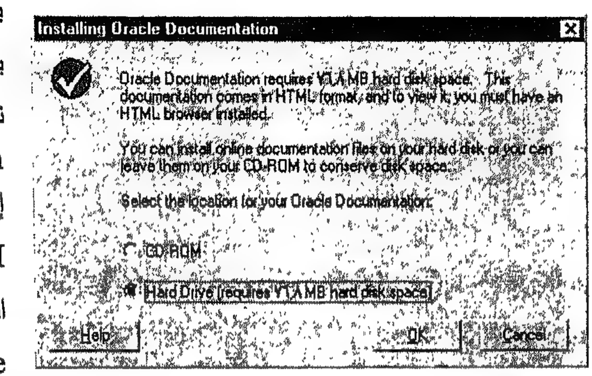
الشكل ٦-٤



الشكل ٦-٥

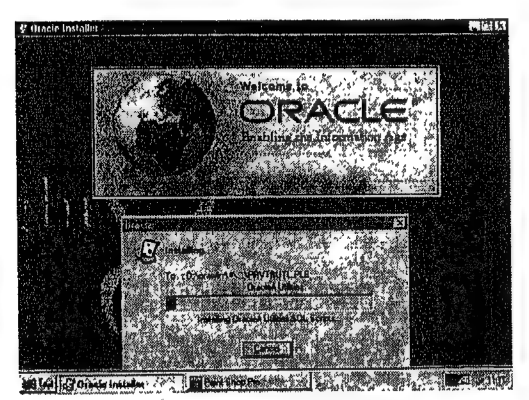
يظهر صندوق حوار جديد يشبه الشكل ٦-٥، يتم فيه تحديد نمط توصيف قاعدة المعطيات التي سيتم إنشاؤها، إما التوصيف النموذجي Typical والذي يتم فيه توليد قاعدة معطيات تلقائياً، أو التوصيف المخصّص Custom والذي تستطيع من خلاله إنشاء قاعدة معطيات وفق التوصيف الذي تقوم بتحديده، أو عدم تحديد أي توصيف، حدد التوصيف المطلوب ثم انقر زر OK.

يظهر صندوق حوار جديد يشبه الشكل ٦-٦، يتم فيه تحديد موقع مستندات أوراكل Oracle Documentation إما على القرص المدمج CD-ROM أو على سواقة القرص الصلب Hard Drive .



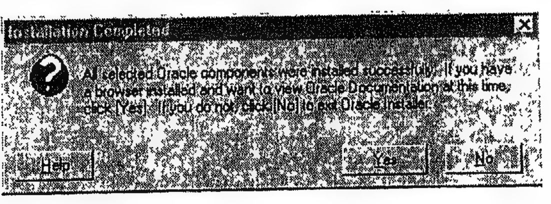
الشكل ٦-٦

حدّد الموقع المطلوب ثمّ انقر زر Ok، تبدأ عملية بدء تركيب الملفات الخاصة بنسخة مخدّم أوراكل وذلك كما في الشكل ٦-٧.



الشكل ٦-٧

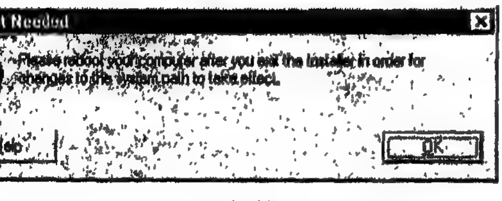
عند الانتهاء من تركيب هذه النسخة بشكل سليم تظهر رسالة توضح أنّه قد تمّ تركيبها بنجاح. وإذا كان لديك مستعرض Browser يمكنك رؤية



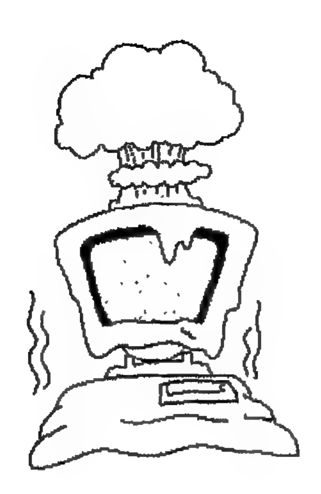
الشكل ٦-٨

مستندات أوراكل بالنقر على زر Yes (انظر الشكل ٦-٨).

تظهر النافذة الأخيرة التي تطلب إعادة إقلاع الحاسوب بعد الخروج من أداة التركيب لأخذ المسارات الجديدة بعين الاعتبار (انظر الشكل ٦-٩).

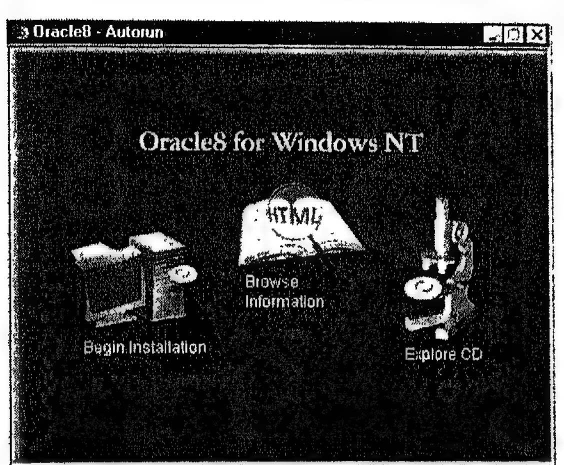


الشكل ٦-٩



تركيب زبون أوراكل Installing Oracle٨ Client

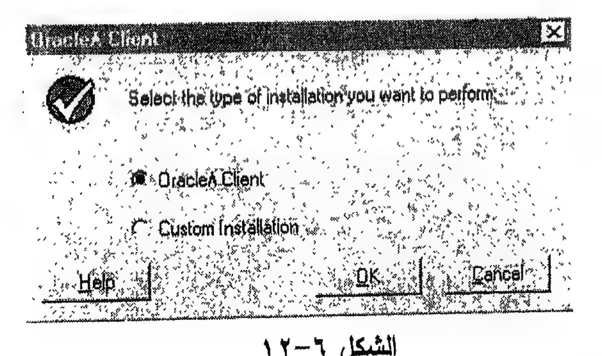
عندما تقوم بوضع القرص المدمج الخاص بنسخة Oracle٨ في سواقة زبون Windows NT Workstation أو زبون Windows٩٥، يتم تلقائياً تشغيل برنامج التركيب كما في الشكل ٦-١٠.



الشكل ٦-١٠

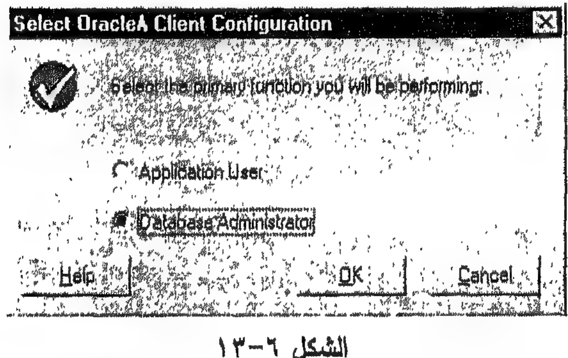
انقر زر Begin Installation لبدء عملية التركيب حيث تظهر نافذة تشبه الشكل ٦-١١، يتم في هذه النافذة تحديد اسم الشركة التي تمتلك النسخة Company Name، كذلك موقع أوراكل Oracle Home وذلك بتحديد اسم الموقع Name، واسم المجلّد الذي سيتم وضع برمجيات أوراكل فيه مع موقع هذا المجلد Location، أخيراً قم بتحديد لغة التطبيق Language.

الشكل ٦-١١



الشكل ٦-١٢

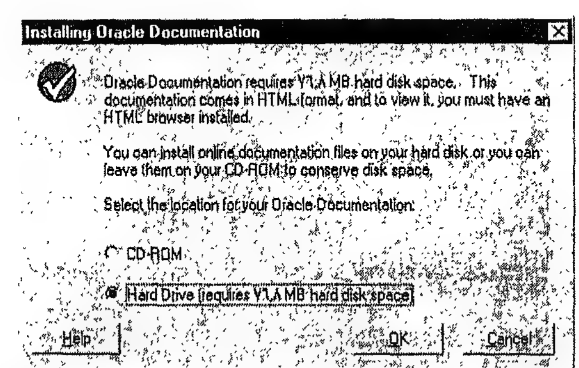
انقر زر Ok يظهر صندوق حوار جديد يشبه الشكل ٦-١٢ لتحديد نمط التركيب، وباعتبار أننا نقوم هنا بتركيب نسخة زبون أوراكل لذلك نختار الخيار الأول Oracle8 Client.



الشكل ٦-١٣

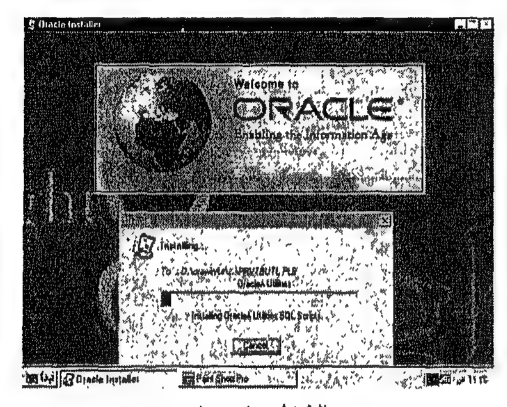
انقر زر Ok يظهر صندوق حوار جديد يشبه الشكل ٦-١٣ يتم فيه تحديد طبيعة عمل الحاسوب الزبون كمستخدم تطبيق فقط Application User لايستطيع أداء أي عملية من عمليات إدارة أوراكل، أو كمدير قاعدة معطيات يستطيع العمل على الحاسوب الزبون تماماً كما لو أنّه يعمل على مخدّم أوراكل. حدد الخيار الثاني حالياً ثم انقر زر Ok.

يظهر صندوق حوار جديد يشبه الشكل ٦-١٤، يتم فيه تحديد موقع مستندات أوراكل Oracle Documentation



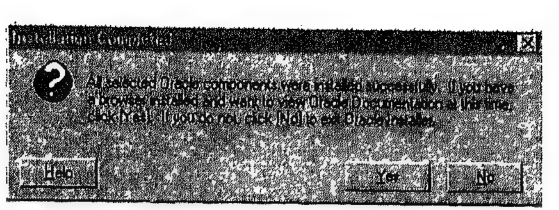
الشكل ٦-١٤

إما على القرص المدمج CD-ROM أو على سواقة القرص الصلب Hard Drive.



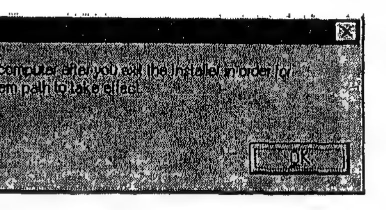
الشكل ٦-١٥

حدّد الموقع المطلوب ثمّ انقر زر Ok، تبدأ عملية بدء تركيب الملفات الخاصة بنسخة زبون أوراكل وذلك كما في الشكل ٦-١٥.



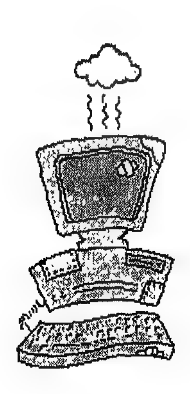
الشكل ٦-١٦

عند الانتهاء من تركيب هذه النسخة بشكل سليم تظهر رسالة توضح أنّه قد تمّ تركيب بنجاح. وإذا كان لديك مستعرض Browser يمكنك رؤية مستندات أوراكل بالنقر على زر Yes (انظر الشكل ٦-١٦).



الشكل ٦-١٧

تظهر النافذة الأخيرة التي تطلب إعادة إقلاع الحاسوب بعد الخروج من أداة التركيب لأخذ المسارات الجديدة بعين الاعتبار (انظر الشكل ٦-١٧).




الباب الثاني

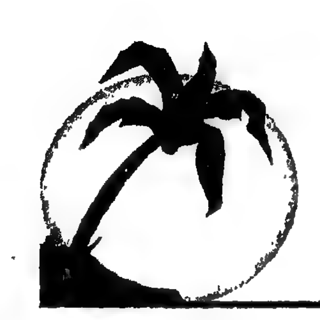


٧. لغة الاستعلام البنيوية *SQL*.

٨. لغة برمجة أوراكل *PL/SQL*.

٩. أوراكل٨ وقواعـــد المعطيـات غرضيـة التوجه.





لغة الاستعلام البنيوية SQL
Structured Query Language

وهي مجموعـــة من الأوامر التي تحتاجها البرامج وكذلك المستخدمين للوصول إلى المعطيات الموجودة ضمن قاعدة معطيات أوراكل.

تـــمّ تطويـــر هـــذه اللغة في البداية من قبل شركة IBM وذلك في منتصف السـبعينات وكـانت تسمى System R حيث كانت عبارة عن نموذج لنظام إدارة قواعد معطيات علاقاتية.

بعدها تمّ توصيف لغة SQL في تشرين ثاني ١٩٧٦ في مجلة IBM Journal of R&D باسـم SEQUEL٢ ومن ثم قامت شركة ORACLE في عام ١٩٧٩ بإنزال أول نسخة تجارية من لغة SQL.

لغة SQL عبارة عن لغة غير إجرائية Non-Procedural Language لأنها تتعامل مع مجموعـة سـجلات فـي الوقت نفسه وليس مع سجل وحيد كما أنها تمكننا من استكشاف المعطيات تلقائياً.

تحـتوي لغـة SQL عـلى تعـليمات تفيد المستخدمين وتمكنهم من إدارة النظام وقواعد المعطيات والتطبيقات كما أنها تحتوي على أوامر لإنجاز مهام مختلفة ومتعددة كالبحث عن المعطيات والتعامل مع قواعد المعطيات والسجلات وتضمن تناسق وتكاملية المعطيات.

لكن مالفرق بين SQL و SQL*Plus؟

كمـا ذكرنا فإنّ SQL عبارة عن لغة تعليمات للاتصال بمخدم أوراكل Oracle Server مـن خلال أية أداة أو أي تطبيق. وعندما تقوم بكتابة تعليمة SQL، يتم تخزينها في جزء مـن الذاكرة يسمى ذاكرة SQL المؤقتة SQL buffer وتبقى فيها حتى تقوم بكتابة تعليمة جديدة.

أما SQL*Plus في عبارة عن أداة من أدوات أوراكل يمكنها التعرف على تعليمات SQL وإرسالها إلى مخدم أوراكل لتنفيذها. وهي تمتلك تعليمات إضافية خاصة بها سنقوم بشرحها لاحقاً.

تسجيل الدخول إلى SQL*Plus

يمكنك القيام بذلك باستخدام أحد الطريقتين التاليتين:

١- من خلال بيئة Windows بطلب الأمر :

*Start -> Programs -> Oracle for Windows NT -> SQL*Plus 8.0*

تظهـر نـافذة تسـجيل الدخول، قم بإدخال اسم المستخدم User Name، وكلمة المرور Password، وسلسلة محارف الحاسب المضيف Host String.

فـي حـال العمل على المخدّم فلا داعي لكتابة سلسلة محارف الحاسب المضيف، أما عند العمـل على الحاسوب الزبون فيجب تحديد اسم الخدم Service (انظر الفصل ١٨ لمزيد من التفاصيل).

٢- من خلال سطر الأوامر Command Line :

تأخذ تعليمة تسجيل الدخول إلى SQL*Plus الشكل التالي :

sqlplus [username[/password[@database]]]



يمكن كتابة التعليمة التالية :

sqlplus scott/tiger@mohib

المعاملات في لغة SQL

توجد مجموعة من المعاملات الأحادية والثنائية في لغة SQL، لهذه المعاملات أفضليـــات موضحة في الجدول التالي :

الأفضلية الأعلى
(معاملات رياضية أحادية) −،+
**،/*
(معاملات ثنائية) ‖،−،+
جميع معاملات المقارنة
Not
AND
OR
الأفضلية الأدنى

تقسم المعاملات إلى الأنواع التالية :

١- **المعاملات الرياضية :** /،*،−،+.

٢- **معاملات المحارف :** ‖ (دمج سلسلتي أحرف).

٣- **معاملات المقارنة :**

NOT IN, IN, >=,<=, >, >, =, IS [NOT] NULL, X[NOT] LIKE Y [ESCAPE Z]EXITS, [NOT] BETWEEN X AND Y, ALL, SOME, ANY

٤- **المعاملات المنطقية:** *OR, AND, NOT*

٥- **معـــاملات المجموعـــات :** *MINUS, INTERSECT, UNION ALL, UNION*

أنماط المعطيات في أوراكل Data types

يبين الجدول التالي ملخص عن الأنماط التي يمكن استخدامها ضمن أوراكل :

النمط	وصفه	طول العمود (بالبايت)
CHAR(size)	نمط حرفي بطول ثابت SIZE.	الحجم الأعظمي ٢٥٥ بايت لكل سطر، الحجم الافتراضي ١ بايت لكل سطر.
VARCHAR2(size)	نمط بطول متغير.	الطول متغير لكل سطر ويمكن أن يصل حتى ٢٠٠٠بايت.
NUMBER(P,S)	نمط معطيات رقمي بطول متغير، دقة أعظمية P وتدرج S.	متغير لكل سطر، السعة العظمى المطلوبة لعمود معطى هي ٢١ بايت لكل سطر.
LONG	نمط معطيات حرفي متغير الطول.	متغير لكل سطر في الجدول ويمكن أن يصل حتى ٢ جيغا بايت.
DATE	نمط معطيات تاريخ ووقت ثابت يتراوح من Jan 1,4712 B.C حتىDec 31,4721 A.C التنسيق الافتراضي DD-MON-YY	٧ بايت لكل سطر.
RAW(SIZE)	نمط معطيات ثنائية بطول متغير يجب تحديده بالحجم SIZE.	متغير لكل سطر في الجدول ويمكن أن يصل حتى ٢٠٠٠ بايت.
LONG RAW	نمط معطيات ثنائية	متغير لكل سطر في الجدول ويمكن

	متغير.	أن يصل حتى ٢جيغا بايت.
ROWIND	نمط معطيات ثنائية تمثل عناوين الأسطر.	ثابت ويساوي ٦ بايت لكل سطر في الجدول.
MLSLABEL	نمط معطيات ثنائية متغيرة الطول تمثل للصيقات LABEL في نظام التشغيل OS.	متغير لكل سطر ويتراوح بين ٢ و٥ بايت لكل سطر.

تعليمات SQL الأساسية

سنقوم هنا بشرح تعليمة SQL الأساسية وهي تعليمة الاختيار SELECT والتي تعتبر أساس هذه اللغة. نستطيع من خلال هذه التعليمة إجراء العمليات التالية :

- الاختيار **Selection** : يمكنك استخدام هذه التعليمة لاختيار أسطر من جدول موافقة لاستعلام ما.
- الإسقاط **Projection** : يتم ذلك من خلال اختيار الأعمدة من الجدول الموافقة لاستعلام ما.
- الربط **Join** : وذلك بالاستعلام عن معطيات موجودة في جداول مختلفة توجد فيما بينها علاقة ارتباط.

تأخذ هذه التعليمة الشكل العام :

```
SELECT      [DISTINCT] {*, column [alias], ...}
FROM        table
[WHERE      condition]
[ORDER BY   {column, expr} [ASC|DESC]]
```

حيث :

- **DISTINCT**: لإلغاء تكرار القيم.
- *: لاختيار جميع الأعمدة.
- **column**: لاختيار أعمدة محددة.

- **alias**: لإعطاء الأعمدة المحددة تسميات أخرى.
- **table**: اسم الجدول.
- **condition**: الشرط الموافق لتعليمة الاختيار.
- **ORDER BY**: لفرز أسطر الاستعلام.
- **ASC**: لاختيار الترتيب التصاعدي لأسطر الاستعلام.
- **DESC**: لاختيار الترتيب التنازلي لأسطر الاستعلام.

لاختيار جميع أعمدة الجدول dept :

```
SELECT    *
FROM      dept;
```

ولاختيار إظهار العمودين deptno و loc من الجدول dept :

```
SELECT    deptno, loc
FROM      dept;
```

يمكنك أيضاً إلغاء تكرار القيم باستخدام عبارة DISTINCT مثلاً :

```
SELECT    DISTINCT deptno
FROM      emp;
```

استخدام المعاملات الرياضية

يمكن استخدام المعاملات الرياضية الأساسية : - ,+ , / ,* فمثلاً يمكن معرفة الراتب السنوي للموظفين بكتابة التعليمة :

```
SELECT    ename, sal, 12*sal "Annual Salary"
FROM      emp;
```

حيث annual_salary هو الرديف للعمود المحسوب 12*sal.

أما لمعرفة الراتب السنوي للموظف KING نكتب:



```
SELECT    ename, sal, 12*sal "Annual Salary"
FROM      emp
WHERE     ename='KING';
```

استخدام معامل الدمج

يمكن استخدام معامل الدمج || على الشكل التالي :

```
SELECT     ename||job AS "Employees"
FROM       emp;
```

يمكننا أيضاً استخدام سلاسل الأحرف ضمن تعليمة SELECT على الشـــكل التالي :

```
SELECT     ename || ' ' || 'is a' || ' ' || job AS "Employees
Details"
FROM       emp;
```

استخدام معاملات المقارنة

يمكنك استخدام معاملات المقارنة التالية : <>, <=, <, >=, >, =. فمثلاً يمكنك معرفـــة أسماء الموظفين الذين تتجاوز رواتبهم 3000:

```
SELECT     ename, sale
FROM       emp
WHERE      sal>3000;
```

كما أنّ بإمكانك استخدام معاملات المقارنة التالية:

- BETWEEN ... AND ....
- IN(list)
- LIKE
- IS NULL

فمثلاً يمكنك معرفة الموظفين الذين تتراوح رواتبهم بين ٣٠٠٠ و ٥٠٠٠:

```
SELECT     ename, sale
FROM       emp
WHERE      sal BETWEEN 3000 AND 5000;
```

كذلك بإمكانك معرفة الموظفين الذيـــن يعملـــون كمديـــري مبيعـــات SALESMAN أو محاسبين CLERK:

```
SELECT     ename, sale
FROM       emp
WHERE      job IN ('SALESMAN', 'CLERK');
```



ولمعرفة الموظفين الذين تبدأ أسماؤهم بالحرف J نكتب:

```
SELECT      ename
FROM        emp
WHERE       ename LIKE 'J%';
```



أخيراً لمعرفة الموظفين الذين لم يحصلوا على كومسيون نكتب:

```
SELECT      ename, sale, comm
FROM        emp
WHERE       comm IS NULL;
```

استخدام المعاملات المنطقية

يمكن استخدام المعاملات المنطقية: AND, OR, NOT ضمن تعليمات الاستعلامات، مثلاً



يمكننا معرفة أسماء المحاسبين الذين تتجاوز رواتبهم ١٠٠٠:

```
SELECT      ename, job, sale
FROM        emp
WHERE       sal>1000 AND job='CLERK';
```

فرز الأسطر

يمكن فرز أسطر نتيجة استعلام تصاعدياً أو تنازلياً وفق عمود ما، فمثلاً يمكننا ترتيب الموظفين تصاعدياً وفق تاريخ مباشرتهم العمل:

```
SELECT      ename, job, sale, hiredate
FROM        emp
ORDER BY    hiredate;
```



ولإجراء الترتيب التنازلي نكتب:

```
SELECT      ename, job, sale, hiredate
FROM        emp
ORDER BY    hiredate DESC;
```

تستطيع أيضاً الفرز وفق عدة أعمدة، فمثلاً يمكن ترتيب الموظفين وفق أرقام أقسامهم وضمن نفس القسم وفقاً لرواتبهم وذلك بشكل تنازلي نكتب:



```
SELECT      ename, deptno, sal
FROM        emp
ORDER BY    deptno, sal DESC;
```

أوامر تحرير SQL*Plus

يمكنك بعد كتابة أوامر SQL إجراء عمليات التحرير عليها باستخدام أوامر SQL*Plus، سنقوم في هذه الفقرة بشرح هذه الأوامر مع إعطاء الأمثلة الموضحة.

أمر الإضافة *Append*

تسمح بإضافة تعليمات إلى نهاية السطر الحالي.

شكل هذه التعليمة :

A[PPEND] text

إذا كتبنا التعليمة التالية :

sql> select ename from

وأردنا إضافة اسم الجدول emp نكتب :

sql> A emp;

وهنا تصبح التعليمة على الشكل :

sql> select ename from emp ;

أمر التعديل *Change*

يسمح هذا الأمر بتغيير كلمة ما ضمن التعليمة بكلمة أخرى.

الشكل العام لهذا الأمر :

sql> C[HANGE] / old / new

أو بحذف نص ما بكتابة الأمر :

sql> C[HANGE] / text /

وإذا لم نضع أي نص جديد بدلاً عن القديم يعني حذفه.



إذا كانت لدينا العبارة التالية :

sql> select enamee from emp dept ;

وأردنا تغيير كلمة enamee إلى ename نكتب الأمر :

sql> c/enamee/ename



ولتكن لدينا مثلاً العبارة التالية:

sql> select ename from emp dept ;

فإذا أردنا حذف كلمة dept نكتب :

sql> c/dept/

فتصبح العبارة بعد الحذف :

sql> select ename from emp;

أمر مسح داريء Clear buffer SQL

كما ذكرنا فإنه يتم تخزين أي تعليمة من تعليمات SQL في ذاكرة مؤقتة تسمى دراىء SQL buffer، يمكننا مسح هذه الذاكرة بكتابة الأمر :

sql> Clear Buffer (cl buff)

لتكن العبارة التالية:

sql> select * from tab;

هذه التعليمة خزنت ضمن الدارئ Buffer (المخزن المؤقت).

وعندما نعطي تعليمة cl buff يقوم بحذف كل الأسطر المخزنة في المخزن المؤقت لـ .sql

أمر حذف الأسطر Delete

يقوم هذا الأمر بحذف الأسطر من التعليمات، وهي تأخذ الشكل:

sql> DEL [n | n m]



فيمكننا مثلاً حذف السطر الحالي بكتابة الأمر:

sql> DEL



ويمكننا أيضاً حذف السطر رقم 2 بكتابة الأمر:

sql> DEL 2



أما لحذف الأسطر من 2 إلى 4 فنكتب الأمر:

sql> DEL 2,4

أمر الإدراج Insert

يفيد هذا الأمر في إدراج عدد غير محدد من الأسطر إلى تعليمة معينة، تأخذ هذه التعليمة الشكل:

sql> I[NPUT] [text]

أمر إظهار مجموعة أسطر من دارئ SQL

الشكل العام لهذا الأمر:

sql> L[IST] [n | n m]

يمكن إظهار السطر الحالي بكتابة الأمر:

sql> L



أما لإظهار السطر رقم 2 نكتب:

sql> L 2

ولإظهار الأسطر من 2 إلى 4 نكتب:

sql> L 2,4

تنفيذ التعليمات Run

يمكن تنفيذ تعليمة SQL الموجودة في ذاكرة SQL المؤقتة بكتابة الأمر:

sql>R[UN]

نقل مؤشر الأسطر

يمكــن نقــل مؤشر الأسطر ضمن التعليمة الموجودة في الذاكرة المؤقتة بكتابة رقم السطر المطلوب الانتقال عليه، مثلاً لجعل السطر ٢ هو السطر الحالي نكتب:

sql>2

ويمكن تبديل هذا السطر بكتابة الأمر:

sql>2 text

يمكنك أيضاً إدراج سطر جديد قبل السطر الأول بكتابة الأمر:

sql>0 text

أوامر الملفات في SQL*Plus

أمر الحفظ *Save*

يمكن حفظ محتوى ذاكرة SQL المؤقتة في ملف بكتابة الأمر:

sql> SAV[E] filename[.ext]

الامتداد الافتراضي للملف هو (.sql)

أمر جلب محتوى ملف *Get*

يســتدعي هذا الأمر محتوى آخر ملف تم حفظه ضمن ذاكرة SQL المؤقتة. الشكل العام لهذا الأمر:

sql> GET filename[.ext]

أمر تنفيذ محتوى ملف *Start*

يمكننا القيام ملف تمّ حفظه من قبل بكتابة الأمر:

sql> STAR[T] filename[.ext]

أو الأمر:

sql> @filename[.ext]

فمثلاً إذا أردنا تنفيذ محتوى الملف mohib.sql نكتب الأمر:



sql> START mohib.sql

أو:

sql> @ mohib.sql

تشغيل برنامج التحرير Edit

يمكن تشغيل المحرر وحفظ محتوى ذاكرة SQL المؤقتة إلى ملف بالاسم afiedt.buf بكتابة الأمر:

sql> ED[IT]

أما لتحرير محتوى ملف ما، نكتب الأمر:

sql> ED[IT] [filename][.ext]

تخزين نتيجة استعلام Spool

يمكننا حفظ نتائج استعلام معين في ملف بكتابة الأمر:

sql> SPO[OL] [filename[.ext]] [OFF|OUT]

حيث :

- OFF : لإغلاق ملف النتائج.
- OUT : لإغلاق ملف النتائج وإرساله إلى الطابعة.

فمثلاً إذا قمنا بكتابة الأوامر التالية:

sql> spool test
sql> select .....
sql> spool off

فإنّه يتم فتح الملف test ووضع نتائج الاستعلام إلى أن يتم إغلاقه بالأمر spool off.

إظهار بنية جدول Describe

تستطيع إظهار بنية جدول بكتابة الأمر:

```
sql> DESC[RIBE] tablename
```



لإظهار بنية الجدول emp نكتب الأمر:

```
sql> DESC emp;
```

استخدام الدالات SQL Functions

تحتوي لغة SQL على العديد من الدالات التي تقوم بإجراء العمليات على المحارف والأرقام والتواريخ وغيرها، وهي تأخذ الشكل العام:

function_name (column|expression, [arg1, arg2, ...])

دالات المحارف Character Functions

هنالك نوعان من دالات المحارف:

١- دالات قلب حالة الأحرف Case Conversion Functions:

الدالة	عملها
LOWER(column|expression)	قلب حالة الأحرف إلى أحرف صغيرة.
UPPER(column|expression)	قلب حالة الأحرف إلى أحرف كبيرة.
INITCAP(column|expression)	قلب الحرف الأول في كل كلمة إلى حرف كبير أما بقية الأحرف فتنقلب إلى أحرف صغيرة.



```
SELECT    empno, ename, deptno
FROM      emp
WHERE     LOWER(ename)='blake';
```

٢- دالات معالجة الأحرف Character Manipulation Functions:

الدالة	عملها
CONCAT(column1|expression1, column2|expression2)	لدمج محارف الوسيط الأول إلى محارف الوسيط الثاني.
SUBSTR(column|expression,m[,n])	لإرجاع المحارف من الموقع m إلى الموقع n.
LENGTH(column|expression)	لإرجاع عدد المحارف.
INSTR(column|expression,m)	لإرجاع موقع حرف محدد.
LPAD(column|expression,n,'string')	لدمج سلسلة المحارف string بالعدد n إلى يسار العمود.
RPAD(column|expression,n,'string')	لدمج سلسلة المحارف string بالعدد n إلى يمين العمود.

```
CONCAT('Hello',' Dear') → Hello Dear
SUBSTR('Hello',1,3) → Hel
LENGTH('Hello') → 5
INSTR('Hello','e') → 2
LPAD(sal,5,'#') → #####5000
```

يوضح المثال التالي كيفية استخدام التوابع السابقة:



```
SELECT    ename, CONCAT(ename,job),
                 LENGTH(ename),
                 INSTR(ename,'A')
FROM      emp
WHERE     SUBSTR(job,1,4)='SALE';
```

دالات الأرقام Number Functions

الدالة	عملها
ROUND(column1|expression1,n)	لتقريب عمود رقمي إلى n خانة عشرية.
TRUNC(column1|expression1,n)	لإرجاع عمود رقمي بعدد محدد n من الخانات العشرية.
MOD(m,n)	لإرجاع باقي قسمة n على m.



ROUND(55.527) → 55.53
TRUNC(55.527) → 55.52
MOD(20,3) → 2



```
SELECT    ename, sal, comm, MOD(sal, comm)
FROM      emp
WHERE     job='SALESMAN';
```

دالات التواريخ Date Functions

الدالة	عملها
MOTHES_BETWEEN	عدد الأشهر بين تاريخين.
ADD_MONTHES	لإضافة شهر إلى تاريخ.
NEXT_DAY	لإعطاء يوم التالي لتاريخ محدد.
LAST_DAY	لإعطاء يوم الأخير في شهر.
ROUND	لطلب تابع ROUND على تاريخ معطى.
TRUNC	لطلب تابع TRUNC على تاريخ معطى.



MONTHS_BETWEEN('01-JAN-99', '01-SEP-99') → 9
ADD_MONTHS('01-JAN-99',9) → '01-SEP-99'
NEXT-DAY('15-MAR-99', 'FRIDAY') → '19-MAR-99'
LAST_DAY('01-MAR-99') → '31-MAR-99'
ROUND('25-MAR-99', 'MONTH') → '01-AVR-99'
TRUNC('25-MAR-99', 'MONTH') → '01-MAR-99'



يوضح الاستعلام التالي كيفية استخدام توابع التاريخ:

```
SELECT    empno, hiredate
          MONTHESBETWEEN(SYSDATE, hiredate)
TENURE,
          ADD_MONTHS(hiredate, 6) REVIEW,
          NEXT_DDAY(hiredate, 'FRIDAY'),
          LAST_DAY(hiredate)
FROM      emp
WHERE     MONTHS_BETWEEN(SYSDATE, hiredate) <100;
```

دالات تغيير أنماط المعطيات

يستطيع أوراكل بشكل تلقائي إجراء عمليات تغيير أنماط المعطيات التالية:

VARCHAR2 or CHAR → NUMBER
VARCHAR2 or CHAR → DATE
NUMBER → VARCHAR2
DATE → VARCHAR2

لكن يزودنا SQL بثلاث دالات لقلب قيمة من نمط معطيات إلى نمط معطيات آخر بشكل خارجي:

الدالة	عملها
TO_CHAR(number|date,['fmt'])	لقلب رقم أو تاريخ إلى VARCHAR2 بتنسيق fmt.
TO_NUMBER(char)	لقلب سلسلة محارف رقمية إلى رقم.
TO_DATE(Char,['fmt'])	لقلب سلسلة محارف تمثل تاريخ إلى قيمة تاريخ بتنسيق fmt.



```
SELECT    empno, TO_CHAR(hiredate, 'DD/MM/YYY')
FROM      emp;
```

دالة NVL

تفيدنا هذه الدالة في قلب قيمة NULL إلى قيمة فعلية موافقة لنمط المعطيات، فمثلاً إذا كتبنا الاستعلام التالي:

```
SELECT    ename,sal, comm, (sal*12)+comm
FROM      emp;
```

ستلاحظ أنه ستظهر فقط الأسطر الموافقة للموظفين الذين حصلوا على كومسيون فقط. لذلك يجب استخدام التابع NVL للحصول على الأسطر الموافقة لكافة الموظفين على الشكل:



```
SELECT    ename,sal, comm, (sal*12)+NVL(comm,0)
FROM      emp;
```

دالة الشرط DECODE

تأخذ هذه الدالة الشكل :

```
DECODE(column|expression, search1, result1
                          [,search2, result2, ... ]
                          [, default])
```

تعمل هذه الدالة على الشكل:

```
IF (column|expression) = search1 THEN result1
ELSE
 IF (column|expression) = search2 THEN result2
 ELSE ....
```

لنفترض مثلاً أننا نرغب بزيادة رواتب الموظفين بحيث يأخذ المدير زيادة ٤٠٪ على راتبه والمحاسب زيادة ٢٠٪ أما المحلل فيأخذ زيادة ١٠٪، يمكننا القيام بذلك بكتابة الاستعلام:

```
SELECT    job, sal
          DECODE(job, 'MANAGER', sal*1.40,
                      'CLERK', sal*1.20,
                      'ANALYST', sal*1.10,
                                 sal) NEW_SALARY
FROM      emp;
```

الاستعلام عن أكثر من جدول

يمكن استخدام الربط Join للاستعلام عن المعطيات الموجودة في أكثر من جدول، يتم ذلك بطلب التعليمة:

```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1=table2.column2;
```

```
SELECT    emp.empno, emp.ename, emp.deptno, dept.loc
FROM      emp, dept
WHERE     emp.deptno = dept.deptno;
```

ويمكن كتابة المثال السابق على الشكل:

```
SELECT    e.empno, e.ename, e.deptno, d.loc
FROM      emp e, dept d
WHERE     e.deptno = d.deptno;
```

تجميع المعطيات

دالات التجميع

يمكن تجميع المعطيات باستخدام دالات التجميع التي تأخذ مجموعات أسطر لإعطاء نتيجة واحدة لكل مجموعة.

يتم استخدام دالات التجميع على الشكل:

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[ORDER BY   column];
```

حيث group_function هي دالة التجميع وهي أحد الدالات التالية:

* AVG([DISTINCT|ALL]n) : دالة المتوسط الحسابي.
* COUNT({*|[DISTINCT|ALL]expr}) : دالة عدد الأسطر.
* MAX([DISTINCT|ALL]expr) : دالة القيمة العظمى.
* MIN([DISTINCT|ALL]expr) : دالة القيمة الصغرى.
* STDDEV([DISTINCT|ALL]x) : دالة الانحراف المعياري.
* SUM([DISTINCT|ALL]n) : دالة المجموع.
* VARIANCE([DISTINCT|ALL]x) : دالة التشتت.



```
SELECT      AVG(sal), MAX(sal), MIN(sal), SUM(sal),
COUNT(sal)
FROM        emp;
```

إنشاء مجموعات معطيات

يمكنك إنشاء مجموعات معطيات باستخدام عبارة GROUP BY على الشكل:

```
SELECT       column, group_function(column)
FROM         table
[WHERE       condition]
[GROUP BY    group_by_expression]
[HAVING      group_condition]
[ORDER BY    column];
```

هنا يجب عليك اتباع مايلي:

- ☆ إذا قمت باستخدام دالة تجميع ضمن عبارة SELECT، فلن تحصل على نتائج إفرادية مالم تظهر الأعمدة الإفرادية ضمن عبارة GROUP BY، وإلا فستحصل على رسالة خطأ.
- ☆ يجب تضمين الأعمدة columns في عبارة GROUP BY.
- ☆ لايمكنك استخدام مرادفات الأعمدة column alias ضمن عبارة GROUP BY.

```
SELECT       deptno, AVG(sal)
FROM         emp
GROUP BY     deptno
ORDER BY     AVG(sal);
```

يمكن تجميع المعطيات على أكثر من عمود، مثلاً:



```
SELECT       deptno, job, SUM(sal)
FROM         emp
GROUP BY     deptno, job;
```



يمكنك استخدام HAVING لإضافة شرط على تابع التجميع، مثلاً:

```
SELECT       job, SUM(sal)
FROM         emp
WHERE        job NOT LIKE 'SALES%'
GROUP BY     job
HAVING       SUM(sal)>300
ORDER BY     SUM(sal);
```

الاستعلامات الفرعية

نحتاج في كثير من الأحيان لدمج استعلامين أو أكثر من أجل حل مسألة معينة، لذلك يمكننا من خلال تعليمات SQL استخدام الاستعلامات الفرعية Subqueries والتي تأخذ الشكل:

```
SELECT      select_list
FROM        table
WHERE       expr operator
                 (SELECT      select_list
                  FROM        table);
```

فمثلاً إذا أردنا معرفة الموظفين الذين يقبضون راتباً أكبر من راتب الموظف 'JONES' نكتب الاستعلام:

```
SELECT      ename
FROM        emp
WHERE       sal >
                 (SELECT      sal
                  FROM        emp
                  WHERE       ename='JONES');
```



كذلك يمكننا معرفة الوظيفة التي لها أقل معدل راتب:

```
SELECT      deptno, AVG(sal)
FROM        emp
GROUP BY    job
HAVING      AVG(sal) =     (SELECT      MIN(AVG(sal))
                            FROM        EMP
                           GROUP BY    job);
```

لغة معالجة المعطيات DML

يتم تنفيذ تعليمات لغة معالجة المعطيات Data Manipulation Language عندما نحتاج إدراج أسطر جديدة إلى جدول، أو تعديل الأسطر الموجودة فيه، أو حذف أسطر منه.

إدراج أسطر جديدة *Insert*

يمكن إدراج أسطر جديدة باستخدام التعليمة التالية:

```
INSERT INTO table[(column [,column...])]
VALUES          (value [,value...]);
```

```
INSERT INTO dept(deptno, dname, loc)
VALUES          (50,          'DEVELOPMENT',
'DAMASCUS');
```

وفي حال رغبنا بإدراج القيم من خلال لوحة المفاتيح نكتب:



```
INSERT INTO dept(deptno, dname, loc)
VALUES          (&department_id,
 '&department_name',
 '&department_location');
```

تستطيع أيضاً نسخ أسطر من جدول آخر، مثلاً إذا رغبنا بنسخ معلومات المديرين من جدول emp إلى الجدول managers نكتب:

```
INSERT INTO managers(id, name, salary, hiredate)
          SELECT    empno, ename, sal, hiredate
          FROM      emp
          WHERE     job='MANAGER';
```

تعديل الأسطر Update

تأخذ تعليمة تعديل الأسطر الشكل:

```
UPDATE     table
SET        column = value [, column = value]
[WHERE     condition];
```

مثلاً إذا أردنا نقل الموظف 'JONES' إلى القسم رقم 20 نكتب:



```
UPDATE     emp
SET        deptno=20
WHERE      ename='JONES';
```

حذف الأسطر Delete

يمكن حذف الأسطر باستخدام التعليمة:

```
DELETE     [FROM]     table
[WHERE                condition];
```



فمثلاً لحذف القسم 50 نكتب:

```
DELETE    FROM    dept
WHERE             deptno=50;
```



لغة تعريف المعطيات DDL

يمكن لقاعدة معطيات أوراكل أن تحتوي بنى معطيات متعددة. فمثلاً يتم تخزين المعطيات في الجداول Tables، ويمكن رؤية مجموعة جزئية من معطيات جدول أو أكثر باستخدام المشاهد Views، أما لتوليد قيم مفتاح أولي فيمكننا استخدام السلاسل Sequences، وباستخدام الفهارس Indexes نستطيع تحسين أداء الاستعلامات.

سنقوم الآن بشرح تعليمات بناء وتعديل وحذف الجداول، أما بقية العناصر فسيتم شرح كيفية التعامل معها في الفصول القادمة.

بناء جدول جديد

يمكن بناء جدول جديد باستخدام التعليمة :

```
CREATE TABLE        [schema.] table
                    (column datatype [DEFAULT expr],...);
```

```
CREATE TABLE        student
                    (sno          NUMBER,
                    name          VARCHAR2(20),
                    address
        VARCHAR2(50));
```

ويمكن بناء جدول بالاعتماد على استعلام فرعي على الشكل:

```
CREATE TABLE        [schema.] table
                    (column (, column...)]
AS                  subquery;
```



مثلاً لبناء جدول جديد يحتوي على موظفي القسم 10 نكتب:

```
CREATE TABLE        dept10
AS
        SELECT      empno,   ename, sal*12 annual_salary,
hiredate
        FROM        emp
        WHERE       deptno = 10;
```

تعديل جدول

يمكن تعديل جدول وإضافة أعمدة جديدة إليه باستخدام التعليمة:

```
ALTER TABLE    table
ADD            (column datatype [DEFAULT expr]
               [, column datatype], ... );
```



```
ALTER TABLE    student
ADD            (school VARCHAR2(20));
```

حذف جدول

يمكن حذف جدول باستخدام التعليمة:

```
DROP TABLE table;
```

لحذف الجدول student نكتب:

```
DROP TABLE student;
```

تغيير اسم جدول

يمكنك تغيير اسم جدول بكتابة التعليمة:

```
RENAME    old_tablename TO new_tablename;
```



```
RENAME    dept10 TO department10;
```

تعريف القيود Constraints

القيود عبارة عن مجموعة من الوظائف الإجبارية التي يتم إنشاؤها على مستوى الجـداول، ويتم إنشاؤها إما في نفس وقت إنشاء الجدول :

```
column [CONSTRAINT constraint_name] constraint_type;
```

أو بعد أن يتم إنشاء الجدول:

```
column,...
        [CONSTRAINT constraint_name] constraint_type
        (column,...),
```

لذلك فإنه يمكن تعريف القيود على الشكل:

```
CREATE TABLE      [schema.]table
                  (column datatype      [DEFAULT expr]
                  [column_constraint],
                  ....
                  [table_constraint]);
```

هنالك العديد من أنواع القيود التي يمكن تعريفها على الجداول أهمها:

- **NOT NULL**: للتحقق من عدم السماح بإدراج قيم معدومة لعمود محدد.
- **UNIQUE key**: للتحقق من عدم تكرار القيم في العمود أو الأعمدة المحددة.
- **PRIMARY KEY**: لإنشاء مفتاح أولي على الجدول.
- **FOREIGN KEY**: لتعريف مفتاح ثانوي على الجدول يرتبط بعلاقة مع مفتاح أولي.
- **CHECK**: لتعريف شرط ما يجب أن يحققه كل سطر في الجدول.

يوضح المثال التالي كيفية تعريف هذه الأنواع من القيود على الجداول:

```
CREATE TABLE employee(
        empno         NUMBER PRIMARY KEY,
        ename         VARCHAR(20) NOT NULL,
        job           VARCHAR2(10) NOT NULL,
        mgr           NUMBER(4),
        sal           NUMBER(7,2) CHECK(sal>1000),
        deptno        NUMBER(7,2) NOT NULL,
        CONSTRAINT          emp_deptno_fk
                            FOREIGN KEY (deptno)
                            REFERENCES dept(deptno));
```

يمكن إضافة قيد جديد باستخدام التعليمة:

```
ALTER TABLE table
ADD [CONSTRAINT constraint] type (column);
```

```
ALTER TABLE                    employee
ADD CONSTARAINT emp_mgr_fk
                               FOREIGN KEY(mgr)
                               REFERENCES employee(emp);
```

أما لحذف قيد من جدول فتستطيع استخدام التعليمة:

```
ALTER TABLE table
DROP PRIMARY KEY | UNIQUE (column) |
        CONSTRAINT constraint [CASCADE];
```

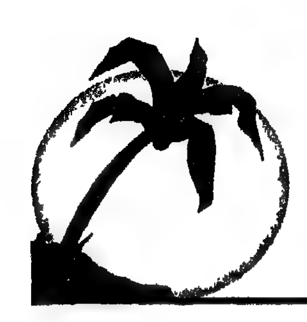


```
ALTER TABLE          employee
DROP CONSTRAINT emp_mgr_fk;
```

أخيراً يمكن تأهيل أو إلغاء تأهيل قيد باستخدام التعليمة:

```
ALTER TABLE table
ENABLE|DISABLE CONSTRAINT constraint [CASCADE]
```





لغة برمجة أوراكل PL/SQL

Oracle Programming Language PL/SQL

كما شاهدنا فإن لغة SQL هي لغة دخول إلى قواعد المعطيات. في الواقع إنـــها لغة مهيكلة وليست مجزأة أي لا يمكن من خلالها أن نعرف دالات وإجرائيات أو أن نكتب برامج لأن البرامج تتطلب حلقات وجملاً شرطية وخلافه.

أما لغة PL/ SQL فهي لغة تدعم لغة الاستعلام البنيوية SQL لتعطيــها إمكانيــة كتابــة برامج متكاملة.

إن الوحدة الأساسية في PL / SQL هي الـكتلة BLOCK والبرنامج هو كتلة مؤلفة مـن كتل جزئية SUB- BLOCKS.

مم يتألف برنامج PL/SQL؟

يتألف برنامج PL/SQL من ثلاثة أقسام:

١- قسم التعريف : وهو قسم اختياري يبدأ بتعليمة DECLARE وينتهي ببداية القسم التالي.

يتم فيه تعريف المتحولات والثوابت، لا يمكن أن ترد في هذا القسم عبارة تنفيذية عدا عبارة الإسناد بالنسبة للثوابت أو تعريف المؤشرات Cursors.

٢- قسم التنفيذ : يبدأ بعبارة BEGIN وينتهي ببداية القسم التالي.

وفيه ترد أوامر لغة PL /SQL أو لغة SQL. يتعامل هذا القسم مع المعطيات الواردة من قاعدة المعطيات. وفي حدوث خطأ نظام أو خطأ معرّف من قبل المستخدم، يمكننا القفز إلى قسم الأخطاء لمعالجتها (القسم الثالث) وذلك باستخدام تعليمة RAISE وهي تأخذ الشكل:

```
IF (condition ) then
Raise value;
```

٣- قسم معالجة الأخطاء : وهو القسم الثالث وينتهي بعبارة END.

ويتم فيه معالجة الأخطاء، وهو يأخذ الشكل العام:

```
EXCEPTION
WHEN value 1
    statement1;
WHEN value 2
    statement2;
WHEN OTHERS
    statementn;
END;
```

يوضح الشكل التالي بنية كتلة PL/SQL:

```
[DECLARE]
        variables, cursors, user-defined exceptions
BEGIN
        SQL statemants
        PL/SQL statements
[EXCEPTION]
        Manipulating Errors
END;
```

يوضح المثال التالي كتلة برنامج PL/SQL:

```
DECLARE
        v_sal   NUMBER;
BEGIN
        SELECT      sal
        INTO        v_sal
        FROM        emp;
EXCEPTION
        WHEN        NO_DATA_FOUND           THEN
            RAISE_APPLICATION_ERROR(-20201,
                            'You don't have a Manager salary');
END;
```

ويمكن أن تكون كتل المعطيات عبارة عن برامج فرعية Subprograms، كالإجرائيــــات Procedures والدالات Functions.

يوضح الشكل التالي بنية كتل البرامج الفرعية:

```
FUNCTION name
RETURN datatype IS
variables, cursors, user-defined
exceptions
BEGIN
        SQL statements
        PL/SQL statements
[EXCEPTION]
        Manipulating Errors
END;
```

```
PROCEDURE name IS
variables, cursors, user-defined
exceptions
BEGIN
        SQL statements
        PL/SQL statements
[EXCEPTION]
        Manipulating Errors
END;
```

المتحولات في لغة PL/SQL

توجد أنماط عديدة من المتحولات أهمها:

- ⌘ المتحولات السلّمية Scalar Variables.
- ⌘ المتحولات المركبة Composite Variables.
- ⌘ المراجع References.
- ⌘ العناصر الكبيرة Large Objects.

ويتم توصيف المتحولات على الشكل:

identifier [CONSTANT] datatype [NOT NULL][:= | DEFAULT expr];

يبين الجدول التالي ملخصاً عن أنماط المتحولات العددية :

النمط	وصفه
CHAR[(maximum_length)]	نمط محارف بطول ثابت يمكن أن يصل حتى 32767، إذا لم يتم تحديد maximum_length تكون القيمة الافتراضية مساوية ١.
VARCHAR2[(maximum_length)]	نمط محارف بطول متغير يمكن أن يصل حتى 32767.
NUMBER(Precision,Scalar)	نمط معطيات رقمي بطول متغير وذلك بدقة أعظمية Precision وتدرج Scalar.
LONG	نمط معطيات محرفي متغير الطول يمكن أن يصل حتى 32767. العرض الأعظم لعمود قاعدة معطيات من هذا النمط حتى 2147483647 بايت.
DATE	نمط معطيات تاريخ ووقت ثابت.
LONG RAW	نمط معطيات ثنائية يمكن أن يصل حتى 32760 بايت.
BOOLEAN	نمط معطيات منطقية.
BINARY_INTEGER	نمط معطيات رقمية بين -2147483647 و 2147483647 +
PLS_INTEGER	يشبه النمط السابق إلا أنه يأخذ مساحة تخزين أقل.

ويمكن استخدام الصفة TYPE% لتوصيف نمط متحول PL/SQL بنفس نمط عمود جدول، مثلاً:

```
v_ename        emp.ename%TYPE;
v_sal          NUMBER(7,3);
v_total_sal    v_sal%TYPE;
```

ويمكن تعريف ثابت بأن نضع قبل النمط كلمة CONSTANT

```
const1 CONSTANT NUMBER (4): =7684
```

عملية إسناد قيمة لمتحول تتم باستخدام معامل الإسناد (=:).

ويمكن أن نسند لمتحول ما نتيجة استعلام باستخدام المحدد INTO مثلاً :



```
SELECT sal INTO var1 from emp where empno= 7654;
```

تعليمات لغة PL/SQL

تمتلك لغة PL/SQL التعليمات الأساسية لأي لغة برمجة كتعليمات الشرط والنسب والحلقات وغيرها.

تعليمات الشرط

وتأخذ الشكل العام:

```
IF condition THEN
        Statements;
[ELSIF condition THEN
        Statements;]
[ELSE
        Statements;]
END IF;
```



```
IF v_sal > 1000 THEN
        v_new_comm := sal*0.20;
END IF;
```

تعليمات الحلقات

توجد العديد من تعليمات الحلقات، كتعليمة الحلقة الأساسية التي تأخذ الشكل:

```
LOOP
        Statements;
EXIT [WHEN condition]
END LOOP;
```

```
LOOP
        INSERT INTO dept(deptno)
                VALUES(v_deptno);
        v_deptno := v_deptno+10;
EXIT WHEN v_deptno>100;
```

أما تعليمة FOR فتأخذ الشكل:

```
FOR counter in [REVERSE]
        lower_bound .. upper_bound LOOP
        Statements;
END LOOP;
```

```
FOR v_deptno IN 50..100 LOOP
INSERT INTO dept(deptno)
                VALUES(v_deptno);
END LOOP;
```

وتأخذ تعليمة WHILE الشكل:

```
WHILE condition LOOP
        Statements;
END LOOP;
```



```
v_deptno := 50;
WHILE v_deptno <100 LOOP
        INSERT INTO dept(deptno)
                VALUES(v_deptno);
        v_deptno := v_deptno+10;
END LOOP;
```

أنماط المعطيات المركبة Composite Datatypes

يمكن تعريف العديد من أنماط المعطيات المركبة في لغة PL/SQL، من أهم هذه الأنماط: السجلات RECORDS، وجداول PL/SQL.

السجلات *Records*

نمط معطيات يمكن أن يحتوي على أنماط فرعية بسيطة أو مركبة، يتم تعريف هذا النمط على الشكل:

```
TYPE type_name IS RECORD
        (field_declaration[, field_declaration]...);
```

ولتعريف متحول من هذا النمط نكتب:

```
identifier type_name;
```



```
TYPE student_record_type IS RECORD
        (st_name        VARCHAR2(20),
        class           NUMBER,
        average         NUMBER(7,2));
st_record       student_record_type;
```



يمكن باستخدام الصفة ROWTYPE% توصيف متحولات بنفس نمط مجموعة أعمدة جدول أو مشهد في قاعدة المعطيات. حيث تأخذ حقول السجل نفس أسماء وأنماط هذه الأعمدة، مثلاً:

```
DECLARE
        employee_record         emp%ROWTYPE;
```

في هذه الحالة يأخذ المتحول employee_record نفس توصيف الجدول emp.

يمكن الوصول إلى أي حقل ضمن متحول سجل باستخدام المعامل (.) وذلك على الشكل:

```
record_name.field_name
```



```
employee_record.sal := 7000;
```

الجداول PL/SQL Tables

يمكن تعريف نمط جدول في لغة PL/SQL، ويتكون هذا الجدول من جزأين:

* المفتاح الأولي لنمط المعطيات BINARY_INTEGER.
* أعمدة ذات أنماط بسيطة أو بنمط سجل.

يتم تعريف أنماط الجداول على الشكل:

```
TYPE type_name IS TABLE OF
        {column_type | variable%TYPE | table_column%TYPE}
[NOT NULL]
        [INDEX BY BINARY INTEGER];
```



```
TYPE ename_table_type IS TABLE OF emp.ename%TYPE
        INDEX BY BINARY_INTEGER;
ename_table ename_table_type;
```

ويتم الوصول إلى أعمدة هذا الجدول على الشكل:

```
pl/sql_table_name(primary_key_value)
```



```
ename_table(1):='LAMIS';
```

المؤشرات SQL Cursor

وهي عبارة عن منطقة عمل SQL خاصة، يوجد نوعان من المؤشرات:

١. المؤشرات الداخلية **Implicit cursors**: يقوم مخدّم أوراكل Oracle Server باستخدام هذا النمط من المؤشرات لعبور وتنفيذ تعليمات SQL الواردة إليه.

٢. المؤشرات الخارجية **Explicit cursors**: ويتم تعريفها من قبل المبرمج.

المؤشرات الداخلية Implicit Cursors

لكل من المؤشرات الداخلية مجموعة من الصفات التي يمكن استخدامها لتفحص نتائج تعليمات SQL، أهم هذه الصفات:

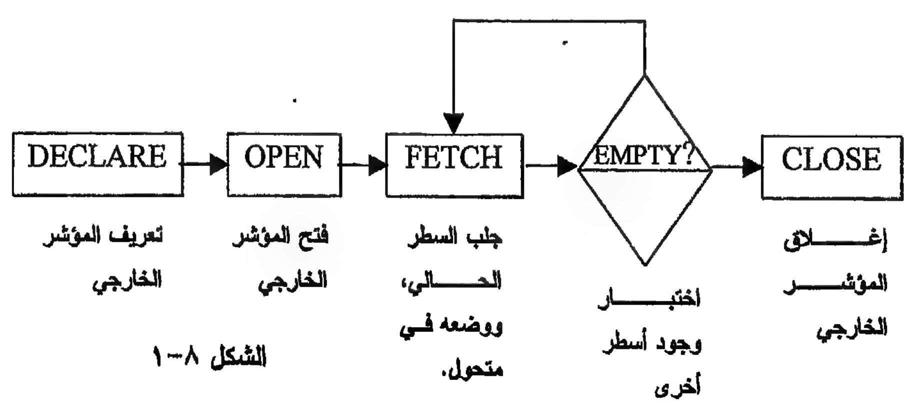
* SQL%ROWCOUNT: عدد الأسطر الناتجة عن تعليمة SQL الجديدة.
* SQL%FOUND: تأخذ قيمة TRUE في حال إيجاد تعليمة SQL لأسطر موافقة، وFALSE في الحالة المعاكسة.
* SQL%NOTFOUND: تأخذ قيمة FALSE في حال إيجاد تعليمة SQL لأسطر موافقة، وTRUE في الحالة المعاكسة.
* SQL%ISOPEN: تأخذ قيمة FALSE دوماً كونه يتم إغلاق المؤشرات الداخلية مباشرة بعد تنفيذها.

المؤشرات الخارجية Explicit Cursors

كما ذكرنا فإنّه يتم استخدام المؤشرات الداخلية من أجل الاستعلامات التي تقوم بإرجاع أكثر من سطر، وذلك من أجل معالجة كل سطر بشكل منفصل.

توجد مجموعة من الخطوات اللازمة للتعامل مع المؤشرات الخارجية، حيث يجب أولاً تعريفها DECLARE ثم فتحها OPEN، بعد ذلك يمكن جلب FETCH أسطر هذه المؤشرات ومعالجتها، أخيراً يجب إغلاقها CLOSE عند انتهاء التعامل معها.

يوضح الشكل ٨-١ آلية عمل هذا النوع من المؤشرات:



الشكل ٨-١

يتم توصيف المؤشر على الشكل:

```
CURSOR cursor_name IS
    select_statement;
```



```
DECLARE
    CURSOR C_d30 IS
        SELECT *
        FROM dept
        WHERE deptno=30;
    ... ...
```

ويتم فتح المؤشر باستخدام التعليمة:

```
OPEN        cursor_name;
```

بينما تستخدم تعليمة FETCH لجلب الأسطر من المؤشر كما يلي:

```
FETCH cursor_name INTO [variable1,variable2,.... | record_name];
```



```
FETCH c_d30 INTO v_deptno,v_dname,v_loc;
```

أما لإغلاق المؤشر فنستخدم التعليمة:

```
CLOSE        cursor_name;
```

لكل من المؤشرات الخارجية مجموعة من الصفات:

- **%ROWCOUNT:** عدد الأسطر الناتجة عن استخدام المؤشر الخارجي.
- **%FOUND:** تأخذ قيمة TRUE في حال تم إرجاع أسطر في آخر عملية جلب FETCH، وFALSE في الحالة المعاكسة.
- **%NOTFOUND:** تأخذ قيمة TRUE في حال عدم إرجاع أسطر في آخر عملية جلب FETCH، وFALSE في الحالة المعاكسة
- **%ISOPEN:** تأخذ قيمة TRUE في حال كون المؤشر مفتوحاً، وFALSE في الحالة المعاكسة.

يمكن استخدام تعليمة حلقة المؤشرات لإجراء عمليات الفتح والجلب والإغلاق عليها، تأخذ هذه التعليمة الشكل:

```
FOR record_name IN cursor_name LOOP
    Statements;
```

```
END LOOP;

DECLARE
        CURSOR c_man IS
                SELECT      empno,sal,job
                FROM        emp
```



```
BEGIN
        FOR emp_record IN c_man LOOP
                IF emp_record.job = 'MANAGER' THEN
                        emp_record.sal := emp_record.sal+2000;
        END LOOP;
END;
```

يمكن استخدام المؤشرات بوسطاء على الشكل التالي:

```
CURSOR cursor_name
        [(parameter_name datatype,...)]
IS
        select_statemant;
```

```
DECLARE
        CURSOR c_no_job(   v_deptno NUMBER,
                           v_job VARCHAR2)
        IS
        SELECT      empno, ename
        FROM        emp
        WHERE       deptno=v_deptno
        AND         job=v_job;
BEGIN
        OPEN c_no_job(20,'ANALYST');
        .....
END;
```

وتستطيع قفل LOCK الأسطر قبل أن تقوم بحذفها أو تعديلها باســـتخدام عبـــارة FOR UPDATE، مثلا:



```
DECLARE
        CURSOR c_dept
                SELECT      deptno,dname
FROM        dept
FOR   UPDATE      NOWAIT;
```



حيث يستخدم الخيار NOWAIT لكي يقوم مخدّم أوراكل بإرجاع خطأ في حال قفل الأسطر من قبل دورة أخرى.

معالجة الاستثناءات EXCEPTIONS

كما ذكرنا سابقاً فإن الاستثناءات عبارة عن محددات في لغة PL/SQL يتم تشغيلها خلال تنفيذ البرنامج، إما بسبب حدوث خطأ أوراكل، أو عند طلبها.

توجد ثلاثة أنواع من الاستثناءات:

* **استثناءات مخدم أوراكل المعرّفة مسبقاً Predefined Oracle Server Exceptions**: لايتم تعريف هذا النوع من الاستثناءات ويتم تشغيلها بشكل داخلي.
* **استثناءات مخدم أوراكل غير المعرّفة مسبقاً Non Predefined Oracle Server Exceptions**: يتم تعريف هذا النوع من الاستثناءات ويتم تشغيلها بشكل داخلي.
* **الاستثناءات المعرّفة من قبل المستخدم User-defined Exceptions**: يتم تعريفها ضمن جزء التوصيف، وتشغّل بشكل خارجي.



استثناءات مخدم أوراكل المعرّفة مسبقاً Predefined Oracle Server Exceptions

يوضح الجدول التالي استثناءات المخدم المعرفّة مسبقاً:

اسم الاستثناء	رقم الخطأ	الوصف
ACCESS_INTO_NULL	ORA-06530	محاولة نسب قيم إلى صفات عنصر غير مصفّر.
COLLECTION_IS_NULL	ORA-06531	محاولة تطبيق مجموعة طرق خلاف EXISTS على جدول متداخل غير مصفّر أو مصفوفة غير مصفّرة.
CURSOR_ALREADY_OPEN	ORA-06511	محاولة فتح مؤشر تمّ فتحه مسبقاً.
DUP_VAL_ON_INDEX	ORA-00001	محاولة إدراج قيمة مضاعفة.
INVALID_CURSOR	ORA-01001	إجراء عملية غير مسموحة على مؤشر.
INVALID_NUMBER	ORA-01722	محاولة فاشلة لقلب سلسلة محارف إلى رقم.
LOGIN_DENIED	ORA-01017	محاولة اتصال فاشلة بأوراكل.
NO_DATA_FOUND	ORA-01403	طلب استعلام لم يرجع أية أسطر.
NOT_LOGGED_ON	ORA-01012	محاولة طلب برنامج PL/SQL دون أن يكون قد تمّ الاتصال بأوراكل.
PROGRAM_ERROR	ORA-06501	خطأ داخلي في برنامج PL/SQL.
ROWTYPE_MISMATCH	ORA-06504	محاولة إجراء عملية نسب غير متوافقة.
STORAGE_ERROR	ORA-06500	إيقاف تنفيذ برنامج PL/SQL يعمل خارج الذاكرة.

SUBSCRIPT_BEYOND_COUNT	ORA-06533	محاولة إرجاع عنصر من جدول متداخل أو مصفوفة برقم فهرس أكبر من عدد العناصر.
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532	محاولة إرجاع عنصر من جدول متداخل أو مصفوفة خارج المجال المسموح.
TIMEOUT_ON_RESOURCE	ORA-00051	انقضاء الزمن بينما يقوم أوراكل بانتظار الحصول على مصدر.
TOO_MANY_ROWS	ORA-01422	تعليمة SELECT بسطر واحد تقوم بإرجاع أكثر من سطر.
VALUE_ERROR	ORA-06502	حدوث أخطاء حسابية أو أخطاء قلب أو دمج أو حجم القيود.
ZERO_DIVIDE	ORA-01476	محاولة التقسيم على الصفر.



استثناءات مخدم أوراكل غير المعرّفة مسبقاً Non Predefined Oracle Server Exceptions

يتم أولاً توصيف هذا النوع من الاستثناءات في قسم التوصيف على الشكل:

```
exception        EXCEPTION;
```

بعدها يجب ربط هذا الاستثناء مع رقم خطأ قياسي ضمن مخدم أوراكل باستخدام التعليمة:

```
PRAGMA        EXCEPTION_INIT(exception, error_number);
```

أخيراً يجب ربط الاستثناء ضمن جزء الاستثناء الموافق.

يوضح المثال التالي كيفية استخدام هذا النوع من الاستثناءات:

```
DECLARE
        e_dept_invalid          EXCEPTION;
        PRAGMA        EXCEPTION_INIT(e_dept_invalid,
-2292);
BEGIN
        ....
EXCEPTION
        WHEN  e_dept_invalid THEN
                message('Departement number in not valid.');
...
END;
```

الاستثناءات المعرّفة من قبل المستخدم User-defined Exceptions

يقوم المستخدم أولاً بتوصيف الاستثناء في جزء التوصيف، على الشكل:

```
exception        EXCEPTION;
```

ويمكن الانتقال إلى هذا الاستثناء من خلال جسم البرنامج الرئيسي باستخدام التعليمة:

```
RAISE        exception;
```



أخيراً يتم تحديد هذا الاستثناء في الجزء الموافق، يوضح المثال التالي كيفية استخدام هذا النوع من الاستثناءات:

```
DECLARE
        e_amount_remaining EXCEPTION;
        ....
BEGIN
....
```

```
RAISE e_amount_remaining;
....
EXCEPTION
        WHEN e_amount_remaining THEN
                message('There is still an amount in stock');
...
END;
```

ويمكن استخدام الإجرائية RAISE_APPLICATION_ERROR لإرجاع رقم خطأ غير قياسي وتجنب حدوث استثناءات غير متداولة. يمكن استخدام هذه الإجرائية في جزء التنفيذ وجزء الاستثناء وهي تأخذ الشكل:

```
RAISE_APPLICATION_ERROR        (error_number,
 message[, {TRUE,FALSE}]);
```



```
EXCEPTION
        WHEN NO_DATA_FOUND THEN
                RAISE_APPLICATION_ERROR(-20201,
                                        'Employee          not
found');
```





أُوراكل٨ وقواعد المعطيات غرضية التوجه

Oracle8 and Object Oriented Databases

سنقوم في هذا الفصل بشرح كيفيّة استخدام ميزات قواعد المعطيـــات غرضيّــة التوجه الموجودة في نسخة Oracle8 من أجل بناء الجيل الجديد من أنظمة قواعد المعطيات غرضية التوجه.

سنتعلم هنا أيضا كيفية استخدام الميزات الأخرى الموجودة في Oracle8 والتي تمكننا مــن إجراء التكامل بين قواعد المعطيات العلاقاتية الموجودة أصلاً وقواعد المعطيات غرضيـــة التوجه، بحيث تستطيع التطبيقات العمل مع هذين النمطين من الأنظمة.

استخدام أنماط العناصر Object Types

من أجل إمكانية دعم تطبيقات قواعد المعطيات غرضية التوجه، يسمح لك Oracle٨ ببناء واستخدام أنماط العناصر Object Types عند تصميم قاعدة المعطيات.



يقوم أوراكل ٨ بدعم جميع خصائص قواعد المعطيات غرضية التوجه عدا الوارثة Inheritance.

```
CREATE OR REPLACE TYPE sales. Part_type As
OBJECT (
Id INTEGER,
Description VARCHAR 2 (50),
On_ hand INTEGER,
Recorder_ point INTEGER,
MEMBER FUNCTION part_ id (descr IN VARCHAR 2 )
RETURN INTEGER,
MEMBER FUNCTION order_part  (part _ id IN INTEGER ,
Quantity IN INTEGER ) );
```

يسمح هذا المثال بالتوضيح السريع لسببين رئيسيين لاستخدام أنماط العناصر. السبب الأول: هو أنها تسمح بتعريف أنماط معطيات معقدة تكون أقرب للواقع الفعلي. والسبب الثاني: هو أن أنماط العناصر تسمح لك بدمج المعطيات والعمليات الموافقة.

إذاً عندما ترغب بإنشاء نمط عنصر جديد، يمكنك توصيف جزأين: توصيف النمط Type Specification، وجسم نمط العنصر Body.

أما الجزء الأول فيتم إنشاؤه باستخدام تعليمة CREATE TYPE، بينما الجزء الثاني فيتم إنشاؤه باستخدام تعليمة CREATE TYPE BODY.

توضح الأمثلة التالية بعض أنماط العناصر وكيفيّة استخدامها عند تصميم قاعدة المعطيات.

المثال الأول يوضح نمط عنصر لتعريف العناوين:



```
CREATE OR REPLACE TYPE pub. Address_ type
AS OBJECT (
Street 1 VARCHAR 2 ( 50 ),
City VARCHAR 2 (50 ),
Country VARCHAR 2 (50 ) );
```

وعلى اعتبار أن النمط السابق لا يحتوي توصيف أي طريقة methode، فإنه لن يحتـوي على جسم نمط body.

الآن عندما تحتاج لتعريف عنوان في جدول قاعدة معطيات علاقاتية، يمكنك استخدام نمـط العناصر السابق ADDRESS_TYPE على الشكل:



```
CREATE TABLE sales. Customers (
Id INTEGER PRIMARY KEY,
Last_ name VARCHAR 2 (50 ),
First_ name VARCHAR 2 (50 ),
Company_ name VARCHAR 2 (50 ),
Address pub_ Address_ type);
```

وعندما ترغب بإجراء عملية استعلام على الجدول الســابق، يمكنـك استخدام التعليمة التالية:



```
SELECT id, last_ name, fist_ name,
Address. Street 1, address. Street 2,
Address. City, address. State,
FROM sales. Customers;
```

يمكنك أيضاً إدراج سجل جديد للجدول السابق كما في المثال التالي:



```
INSERT INTO sales. Customers VALUES (
1, 'NOUKARI', 'Mohib', Oracle agency',
pubm Address_ type (
'Abo_ Romaneh', 'DAMASCUS', 'SYRIA' ) );
```

استخدام الجداول المتداخلة Nested Tables

من الإمكانيات الجديدة التي يقدّمها Oracle٨، إمكانية إنشاء جدول ضمن جدول آخر باستخدام أنماط العناصر.

ويستطيع أوراكل تلقائياً إدارة علاقة الارتباط بين كل سطر من الجدول الأب والأسطر المرتبطة به من الجدول الابن.

ويفيدنا هذا النوع من الجداول في إزالة تعقيد علاقات الربط relational joins من التطبيقات.

كذلك فإن الجداول المتداخلة مناسبة جداً لعلاقات السيد/العبد master/detail، فلنفترض مثلاً أن لدينا جدولاً ITEMS يمثل عناصر شركة ما، و ORDERS جدولاً آخر يمثل طلبات الشراء ويرتبط هذان الجدولان بعلاقة السيد/العبد، نستطيع إنشاء النمط ITEM_ TYPE (لا يحتوي على أي مرجع للحقل ID) والنمط الموافق ITEM_ LIST واللذان يمكننا استخدامهما لإجراء التداخل بين الجدولين ORDERS, ITEMS:



```
CREATE OR REPLACE TYPE sales. Item_ type
AS OBJECT (
Item_ id INTEGER,
Quantity INTEGER);
CREATE OR REPLASE TYPE sales. Item_ list
AS TABLE OF sales. Item_ type;
CREATE TABLE sales. Orders (
Id INTEGER PRIMARY KEY,
Order_ date DATE,
Ship_ date DATE,
Line_ items sales. Itemes STORE AS items;
```

وفي حالة الجداول المتداخلة فإن أوراكل يقوم بتخزين المعطيات في شريحة معطيات فيزيائية وحيدة، لكن يقوم بإنشاء جدولين منطقيين في قاموس المعطيات.

يمكنك إجراء عملية إدراج للمعطيات بسهولة في المثال السابق مثلاً:

```
INSERT INTO sales. Orders VALUES (
1, SYSDATE, NULL,
sales. Item_ List (
sales. Item_ type ( 1,22 ),
sales. Item_ type ( 2,100 ) ) );
```

يمكنك أيضاً استخدام الاستعلامات الفرعيّة الممهّدة flattened subquery والمحدّدة بتعبير SQL الخاص THE، يبين المثال التالي كيفية إدراج عنصر جديد للطلبية رقم ١:



```
INSERT INTO THE (SELECT line_ itemes
FROM sales. Orders WHERE id = 1 )
VALUES (3, 200 );
```

تستطيع أيضاً إجراء عملية البحث في الجداول المتداخلة باستخدام الاستعلامات الفرعيّة الممهّدة flattend subquery، مثلاً:



```
SELECT item_ id, quantity
FROM THE ( SELECT line_ items FROM sales. Orders
WHERE id =1 )
ORDER BY item_ id;
```

أما لحذف الأسطر من الجداول المتداخلة، يمكننا مثلاً كتابة:



```
DELETE THE ( SELECT line_ items FROM
Sales. Orders WHERE id = 1 ) n
WHERE n .item_ id = 3;
```

إنشاء عناصر الجداول Creating Object Tables

عنصر الجدول Object table عبارة عن جدول قاعدة معطيات يتم بناؤه باستخدام أنماط العناصر فقط Object types وليس أعمدة العلاقات relational column.

وعندما تقوم بإنشاء عنصر جدول، فإن أعمدة الجدول تكون عبارة عن واصفات attributes نمط العنصر الذي تمّ استخدامه لبناء الجدول.

أما أسطر الجدول فهي عبارة عن عناصر من نمط الجدول، لكل عنصر محدد عنصر وحيد Object Identifier (OID)، ويقوم أوراكل باستخدام محددات العناصر OIDs لتعريف علاقات الارتباط relationships بين مختلف عناصر الجداول في قاعدة المعطيات.

يبين المثال التالي كيفية بناء مخطط مبيعات مواد قطع حواسيب باستخدام أنماط العناصر object types وعناصر الجداول object tables:



```
-- Statements to create a CUSTOMER object table
CREATE OR REPLACE TYPE sales. Customer_ type AS OBJECT (
Id INTEGER,
Last_ name VARCHAR 2 (50),
First_ name VARCHAR2 (50),
Company_ name VARCHAR 2 (50),
Address pub. Address_ type );
CREATE TABLE sales. Customer OF sales. Customer_ type
(id PRIMARY KEY);

--Statements to create a PARTS object table
CREATE OR REPLACE TYPE sales. Part_ type As OBJECT (
Id INTEGER,
Description VARCHAR 2 (50 ),
Unit_price NUMBER (10, 2 ),
On_ hand INTEGER,
Recorder_point INTEGER );
CREATE TABLE sales. Parts OF sales. Part_ type
(id PRIMARY KEY );
-- Statements to create on ORDER object table with a nested ITEM
table
CREATE OR REPLACE TYPE sales. Item_ type AS OBJECT (
Item_ id INTEGER,
Part REF sales. Part_ type,
Quantity INTEGER);
CREATE OR REPLACE TYPE sales. Item_ list AS TABLE or sales.
Item_ type;
CREATE OR REPLACE TYPE sales. Order_ type AS OBJECT (
Id INTEGER,
Customer REF sales. Customer_ type,
Order_ date DATE,
Ship_ date DATE,
Line_ items sales. Item_ list );
CREATE TABLE sales. Order OF sales. Order_ type
( id PRIMARY KEY)
NESTED TABLE line_ items STORE AS items.
```

الآن لنرى كيف نستطيع التعامل مع عناصر الجداول باستخدام تعليمات SQL.



يوضــح المــثال الــتالي كيفيــة استخدام طرق البناء Constructor methods لإدراج بعــض المعطيات إلى جدولي CUSTOMERS, PARTS:

INSERT INTO sales. Parts
VALUES (sales. Part_ type (1, 'pentium 450 CPU',
250, 1000, 300));
INSERT INTO sales. Customers
VALUES (sales. Customer_ type (1, 'NOUKARI', 'MOHIB', 'Oracle Agency',
Pub. Address_ type ('Abo_ Romanh', 'DAMASCUS', 'SYRIA')));

وعــندما يحــتوي عنصــر جدول على واصف attribute عبارة مرجع عنصر object reference فإنه يتم استخدام التابع REF الذي يقوم بإعادة مرجع أو مؤشر pointer إلى محدد عنصر OID لعنصر خاص ويمكن إيضاح ذلك بالمثال التالي:

INSERT INTO sales. Orders
SELECT 1, REF (e), SYSDATE, NULL, sales. Item_ List ()
FROM sales,. Customers c
WHERE id = 1;

الآن لاستكمال عناصر سطر الفاتورة، يجب استخدام تعليمات INSERT مع الاستعلامات الفرعيّة الممهّدة flattened subqueries كما يوضح المثال التالي:

INSERT INTO THE (SELECT o. line_ items FROM Orders O   WHERE O. id = 1)
SELECT 1 REF (p), 20 FROM parts p
WHERE id = 2;

يوضــح المــثال الــتالي كيــف يمكن لتطبيق قاعدة معطيات استخدام مــتحولات الــبرنامج لإرجــاع محــددات العناصر OIDs ضمن تعليمات SELECT و INSERT و UPDATE:



DECLARE
- - declare variables to hold OID references
custoid REF sales. Customer_ type;
partoid REF sales, part_ type;
BEGIN
- assign Amer Saed OID to CUSTOID
SELECT REF (c) INTO custoid FROM sales. Customers c,
WHERE c. last name = 'Saed'

```
AND c. first_ name = 'Amer';
- assign Pentium 450', OID to PARTOID
SELECT REF (p ) INTO partiod FROM sales. Parts p
WHERE p. description = 'pentium 450 CPU';
 --insert anew order for CUSTOID the order has on
- line item for PARTOID
- assign the new order,s OID to ORDIOD
INSERT INTO sales. Orders
VALUES ( sales. Order_ type ( 1, custoid, sysdate, NULL, sales. Item_
List ( sales. Item_ type ( 1, Partoid, 50 ) ) );
EXCEPTION
WHERE NO_ DATA_ FOUND THEN
Raise_ application_ error (20000, 'No data found' );
END;
```



سيخيّل إليك في هذه المرحلة بأن الخطوات اللازمة لإدخال المعطيات إلى عناصر الجداول صعبة ومعقدة مقارنة مع مثيلاتها في تعليمات العلاقاتية. لكن تذكر بأنك تحتاج إلى بعض التمرين للتعود على استخدامها.

تذكّر أيضاً أن جداول العناصر تسهّل عليك كثيراً بناء الاستعلامات باستخدام محدّدات العناصر OIDs خاصةً عندما ترغب بدمج المعلومات بين جداول العناصر المرتبطة، بدلاً من بناء استعلامات ربط معقدة في النموذج العلاقاتي.

ويقوم Oracle٨ تلقائياً بالتنقل بين مراجع العناصر لجعل ترميز SQL أكثر وضوحاً.

لنأخذ المثال التالي الذي يعطينا المعلومات المرتبطة بين جدولي العناصر ORDERS و CUSTOMERS:

```
SELECT o. id , o. customer, company_ name
FROM sales. Orders o;
```

بينما في النموذج العلاقاتي، فإنك تحتاج إلى فهم علاقة الارتباط بين جدولي CUSTOMERS, ORDERS ومن ثم ترميز هذه العلاقة في كل استعلام لربط المعلومات بين هذين الجدولين كما يوضح المثال التالي:

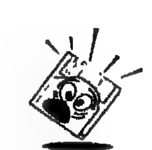


```
SELECT o. id, c. company_ name
FROM sales. Orders o, sales. Customers c
WHERE o. cust_ id = c. id;
```

استخدام الطرق Using Methods

عندما تقوم بتوصيف الطرق ضمن جزء توصيف نمط العنصر فإنه يتوجّب عليك تحديد هذه الطرق بإنشاء الترميز الموافق في جسم نمط العنصر. ويتم ذلك باستخدام تعليمة CREATE TYPE BODY وتعليمات لغة برمجة PL/SQL.

لذلك فإن الطرق methods تشبه كثيراً إجرائيات أو توابع PL/SQL التي رأيناها سابقاً والتي يتم تخزينها ضمن نمط العنصر ويتم تضمينها encapsulate في النمط.



ضمن النسخة الحالية من أوراكل Oracle8 فإن الطرق لا تقوم بشكل فعلي بتضمين Encapsulate أنماط العناصر، وذلك لأن التطبيقات يمكنها استخدام عناصر SQL للوصول إلى عناصر نمط بدلاً من طرق نمط العنصر الموافق.

يمكن لنمط العنصر أن يأخذ أنماطاً مختلفة من الطرق كطرق البناء constructor method وطرق الأعضاء member method وطرق الخريطة map method أو طرق الترتيب order method.

طرق البناء constructor method

يقوم أوراكل تلقائياً بإنشاء طريقة بناء لنمط عنصر، لذلك يمكنك توليد عناصر جديدة لنمط جديد. ويأخذ باني النمط افتراضياً نفس الاسم الذي يأخذه النمط نفسه أما وسطاؤه parameters فتأخذ واصفات attributes نمط العنصر.

توضح الأمثلة السابقة في هذا الفصل طرق بناء أنماط العناصر.

طرق الأعضاء member method

يمكن لكل نمط عنصر أن يمتلك طريقة أو أكثر من طرق الأعضاء، وليست هذه الطرق سوى إجرائيات مخزنة أو توابع ترتبط بنمط العنصر نفسه.

ولتفادي الآثار الجانبية غير المرغوبة، فإن طرق أنماط العناصر لا تمتلك حق إدراج أو حذف أو تعديل المعلومات في جداول قواعد المعطيات.

الخطــوة الأولى والضرورية لإنشاء أنماط الأعضاء هي توصيفها كجزء من توصيف نمط العنصـــر. يوضح المثال التالي كيفية إنشاء توصيف ORDER, TYPE كنمط عنصر مع طريقة عضو والجدول ORDERS الموافق:

```
CREATE OR REPLACE TYPE sales. Order_ type AS
OBJECT (
Id INTEGER,
Customer REF sales, customer_ type,
Order_ date DATE,
Ship_ date DATE,
Line_ items sales. Item_ list,
MEMBER FUNCTION order_ total RETURN NUMBER,
PRAGMA RESTRICT_ REFERENCES ( oder_total, WNDS, WNPS ) );
CREATE TABLE sales. Order_ type,
( id PRIMARY KEY )
NESTED TABLE line_ items STORE AS items;
```

من المثال السابق يمكنك ملاحظة مايلي:

- يمكن لكل طريقة أن تمتلك وسيطاً أو أكثر، وربما لا تمتلك أي منها.
- يجب على تابع الطريقة member function إرجاع قيمة واحدة فقط.
- يجــب على كل طريقة أن تمتلك مترجم PRAGMA والذي يحدد العمليات التي تســتطيع الطــريقة إجراءها. في المثال السابق يمكن للطريقة ORDER_TOTAL إجــراء WNDS (Write No Database State) كذلــك إجــراء (Write No Package State), WNPS.

أمــا بقيــة العمــليات فهــي RNDS (Read No Database State) و (Read No Package State) RNPS.

أما لتحديد محتوى الطرق السابقة فيمكن رؤية المثال التالي:



```
CREATE OR REPLACE TYPE BODY sales. Order_ type (
MEMBER FUNCTION order_ total RETURN NUMBER
IS return_ value NUMBER,
BEGIN
SELECT SUM ( lquantity * l. part_ unit_price )
INTO return_ value
FROM THE ( SELECT o. line_ items FROM sales. Orders,
WHERE o, id = SELF. Id ) l;
RETURN return  value;
```

```
END order_total;
 );
```



يمكنك الآن استخدام الطريقة order_ total بسهولة ضمن تعليمات SQL مثلاً:

```
SELECT o. order_ total () FROM sales. Orders o
WHERE id =1;
```

قارن الطريقة السابقة مع الاستعلام العلاقاتي الذي يقوم بنفس الغرض والذي يمكن كتابتــه على الشكل:



```
SELECT SUM ( i. Quantity * P. unit_ price )
FROM sales. Items I. Sales. Parts p
WHERE i. Order_ id = 1
AND i. Part_ id = p. id;
```

طرق الترتيب والخريطة Order and Map Methods

يمكن لأوراكل بسهولة إجراء المقارنة بين المعطيات في الأعمدة ذات الأنماط القياسية مثل: NUMBER, DATE, CHAR. ويقوم أوراكل باســتخدام Map method أو Order Method، لإجراء العمليات التالية:

- علاقات المساواة والأكبر و الأصغر من.
- عبارات IN, BETWEEN.
- عبارات DISTINCT, GROUP BY, ORDER BY.
- مقيدات PRIMARY KEY, UNIQUE.

يبين المثال التالي توصيف نمط العنصــر ADDRESS_ TYPE الــذي يحتوي على map method:

```
CREATE OR REPLACE TYPE pub. Orders_ type AS
OBJECT (
Street 1 VARCHAR 2 ( 50 ),
Street 2 VARCHAR 2 (50),
City VARCHAR 2 (50),
State VARCHAR 2 (25),
Zipcode VARCHAR 2 (10),
Country VARCHAR 2 (50),
MAP MEMBER FUNCTION address_ map RETURN VARCHAR 2,
```

```
);
CREATE OR REPLACE TYPE BODY address_ type (
MAP MEMBER FUNCTION ON address_ map RETURN VARCHAR 2
IS
BEGIN
  RETURN Zipcode \\ city \\ street 1;
END address_ map;
```

الآن عندما يطلب من استعلام ترتيب السجلات وفق العناوين، يقوم أوراكل تلقائياً باستخدام طريقة ADDRESS_ TYPE لإجراء عملية الترتيب:

```
SELECT c. company. Name, c. address. Zepcode
FROM sales. Customers c
ORDER BY c. address;
```

يمكنك بناء طريقة ترتيب order method لإجراء نفس العمل ولكن بشكل أصعب قليلاً:

```
CREATE OR REPLACE TYPE pub. Address_ type AS
OBJECT (
Street 1 VARCHAR 2 (50);
....
ORDER MEMBER FUNCTION address_ map ( other address_ type )
RETURN INTEGER
 );
CREATE OR REPLACE TYPE  BODY pub. Address_ type (
ORDER MEMBER FUNCTION address_ map ( other address_ type )
RETURN INTEGER IS
Self_ address VARCHAR 2 (50) : = self. Zecode \\
                                             Self. City \\
                                              Self. Street 1;
Other_ address VARCHAR 2 (150) : = other. Zecode \\
                                                  Other. City \\
                                                   Other street 1;
BEGIN
IF self_ address < other_ address THEN
RETURN – 1;
ELSIF self_ address > other_ address THEN
RETURN 1;
ELSE
RETURN 0;
END IF;
```

```
END address_ map;
```

استخدام مشاهد العنصر Using Object Views

يمكـنك إنشـاء مشـهد عنصر object view لأي عنصر جدول Object table، فمثلاً المشـهد الـتالي لعنصـر الجـدول CUSTOMERS يسـتخدم النمط CUSTOMER_ TYPE لوصف بنيته:

```
CREATE OR REPLACE VIEW sales. Cust OF sales.
Customer_ type AS
SELECT * FROM sales. Customers;
```

ويمكـن ضـمن Oracle٨ تعديل مشهد عندما تقوم بإنشاء إجراء INSTEAD OF لهذا المشهد. يقوم هذا الإجراء بإخبار أوراكل كيفية تطبيق تعليمات لغة إدارة المعطيات DML على المشهد.



فمـثلاً إجـراء INSTEAD OF التالي يحدّد ما الذي يجب عمله عندما يتلقى تطبيق تعليمة INSERT ضمن المشهد ORD:

```
CREATE OR REPLACE TRIGGER ord_ insert_ trigger
INSTEAD OF INSERT ON sales. Ord
DECLARE
Item_ var sales. Item_ list;
 I INTEGER
Cust_ var sales. Customer_ type;
Part_ var sales. Part_ type;
Part_ var_ ref REF sales. Part_ type;
BEGIN
Item_ var : = new. Line_ items;
SELECT DEREF ( : new. Customer ) INTO cust_ var FROM dual;
INSERT INTO sales. Orders.
VALUES ( : new. Id, cust_ var. id, : new. Order_ date,
: new. Ship_ date);
FOR I IN 1 .. item_ var. count LOOP
Part_ var_ ref : = item_ var ( I ) . part;
SELECT DEREF ( part_ var_ ref ) INTO
Part_ var FROM dual;
INSERT INTO sales. Items
```

```
VALUES ( new. Id, part_ var. id,
Item_ var ( I ). Item_ id,
Item_ var ( I ). Quantity );
END LOOP,
END;
```

حيث يقوم التابع DEREF بإرجاع قيمة معطيات العنصر المرجع.

الآن يستطيع التطبيق إدراج عناصر جديدة في الجدولين ITEMS, ORDERS باستخدام تعليمات SQL الخاصة بالعناصر، يمكننا إظهار ذلك عن طريق المثال التالي:



```
INSERT INTO sales. Ord
SELECT 1, REF ( c), SYSDATE, NULL, sales. Item_list ( )
FROM sales. Cust c
WHERE id = 1;
```




الباب الثالث



١٠. البنية الفيزيائية لقاعدة معطيات أوراكل.

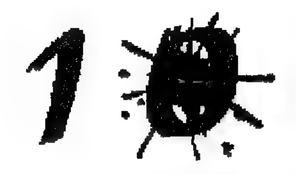
١١. البنية المنطقية لقاعدة معطيات أوراكل.

١٢. بنى ذاكرة أوراكل.

١٣. بنى إجراءات أوراكل.

١٤. بنى إضافية.

١٥. قاموس المعطيات.





البنية الفيزيائية لقاعدة معطيات أوراكل
Database Physical Structure

تتكون قاعدة معطيات أوراكل من مجموعة من الملفات الأساسية التالية :

- ملفات المعطيات **Data files**.
- ملفات الإرجاع **Redo log files**.
- ملفات التحكم **Control files**.

ملفات المعطيات Data files

تتكون قاعدة معطيات أوراكل من ملف أو أكثر من ملفات المعطيات، تحتوي ملفات المعطيات على جميع معطيات القاعدة.

ويتم تخزين معطيات بنى قواعد المعطيات المنطقية كالجداول والفهارس بشكل فيزيائي ضمن ملفات المعطيات.

تتميز ملفات المعطيات بما يلي :

- يمكن ربط ملف معطيات مع قاعدة معطيات واحدة فقط.
- يمكن لملف أو عدّة ملفات معطيات تكوين وحدة تخزين منطقية اسمها الفضاء الجدولي Tablespace.

عندما تتم قراءة معطيات من ملف معطيات يتم تخزينها في الذاكرة المخبئية Cache memory لأوراكل. وبالتالي عندما يحتاج مستخدم إلى معلومات غير موجودة في الذاكرة المخبئية تتم قراءتها من ملف المعطيات المطلوب.

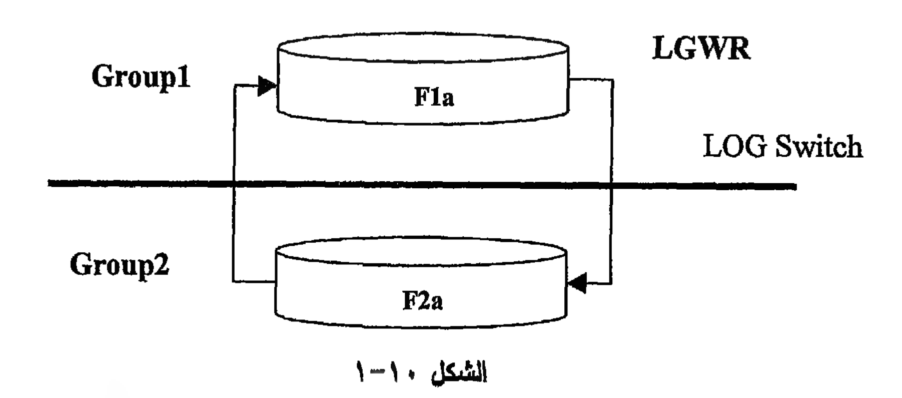
وليس ضرورياً أن تتم كتابة المعطيات الجديدة أو المعدّلة مباشرةً في ملف المعطيات، وإنما يتم تجميع المعطيات في الذاكرة ومن ثم كتابتها في ملف المعطيات دفعةً واحدة. طبعاً هذا يقلل من عمليات الكتابة على القرص.



ملفات الإرجاع Redo log Files

تحتوي أية قاعدة معطيات ضمن أوراكل على ملفين أو أكثر من ملفات الإرجاع. الهدف الرئيسي من هذه الملفات تسجيل جميع التغييرات التي يتم إجراؤها على المعطيات. بالتالي عند حدوث أي عطل يمنع من كتابة المعطيات على ملفات المعطيات كحدوث انقطاع كهرباء مفاجئ، يمكن الحصول على التغييرات التي تمّ القيام بها من ملفات الإرجاع وعدم ضياع العمل الذي تمّ إجراؤه.

الحالة الأبسط لاستخدام ملفات الإرجاع هي استخدام ملفي إرجاع بحيث تتم كتابة معلومات الإرجاع في الملف الأول، وعندما يمتلئ هذا الملف يتم متابعة الكتابة في الملف الثاني وهكذا... (انظر الشكل ١٠-١):



الشكل ١٠-١



ملفات التحكم Control Files

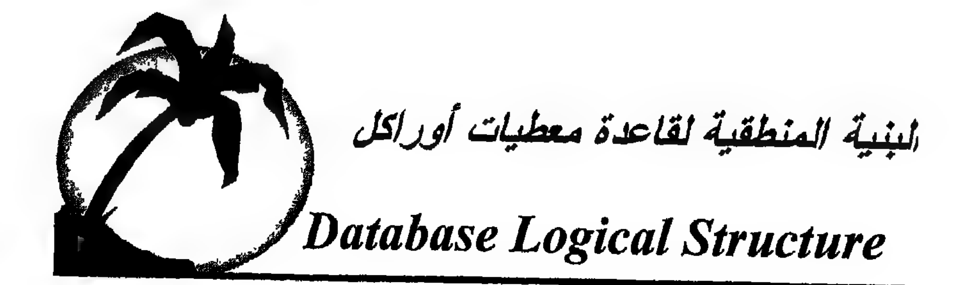
لكل قاعدة معطيات أوراكل ملف تحكم واحد على الأقل يحتوي على معلومات عن البنية الفيزيائية للقاعدة، يمكن أن يحتوي هذا الملف على المعلومات التالية :

- اسم قاعدة المعطيات.
- أسماء وأماكن ملفات المعطيات وملفات الإرجاع.
- وقت إنشاء القاعدة.

ويستطيع نظام أوراكل إنشاء نسخة مماثلة من ملف التحكم لحمايته.



11

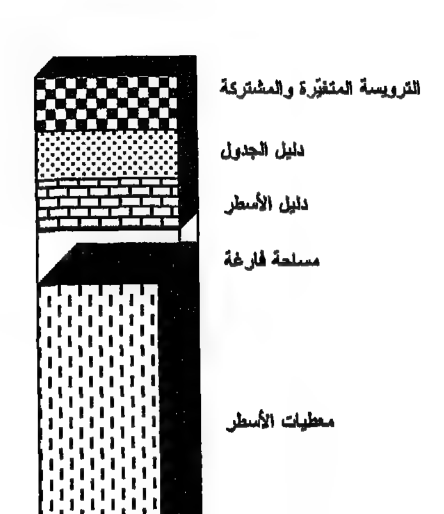


البنية المنطقية لقاعدة معطيات أوراكل
Database Logical Structure

تتكون قاعدة معطيات أوراكل من مجموعة من البنى المنطقية هي:

- كتل المعطيات **Data Blocks**.
- المدى **Extents**.
- المقاطع **Segments**.
- الفضاءات الجدولية **Tablespaces**.

كتل المعطيات Data blocks



الشكل ١١-١

تعتبر أصغر وحدة منطقية، وهي تمثل عدداً من البايتات على قرص التخزين ويمكن تحديد حجم كتلة المعطيات data block size عند إنشاء قاعدة المعطيات.

تتألف كتلة المعطيات من الأقسام التالية (انظر الشكل ١١-١):

١- الترويسة المتغيرة والمشتركة **Common and variable header**: وهي تتضمن عنوان الكتلة ونوع المعطيات داخلها، وحجمها يتراوح بين ٨٤ - ١٠٧ بايت.

٢- دليل الجدول **Table Directory**: ويحتوي أسماء الجداول والتي لها معلومات ضمنه.

٣- دليل الأسطر **Row Directory**: يحوي معلومات عن الأسطر الموجودة داخله.

٤- القسم الفارغ **Free Space**: يراعى دائماً ترك قسم فارغ داخل كتلة المعطيات من أجل التعديل فيما بعد. حجم هذا القسم يحدد بمعاملين هما PCTFREE وPCTUSED وسندرسهما بالتفصيل لاحقاً.

٥- معطيات الأسطر **Row Data**: في هذا القسم تخزن المعلومات المدخلة للجدول.

(يمكن أن تكون كتلة المعطيات بحجم 2 أو 4 أو 8 كيلو بايت حسب حجم القاعدة).

المدى Extent

وهو المستوى الأعلى من كتل المعطيات في أوراكل، ويمثل عدداً محدداً من كتل المعطيات المتجاورة Contiguous data blocks والتي تستخدم في تخزين نمط معين من المعلومات.

المقاطع Segments

تقسم المقاطع إلى أربعة أنماط :

١. **مقاطع المعطيات data segments** : لكل جدول غير مجمّع مقطع معطيات خاص به، ويتم وضع معطيات جدول في المدى الخاص بمقطع المعطيات. كذلك لكل تجمّع مقطع معطيات خاص به، حيث يتم تخزين كل جدول من جداول التجمّع في مقطع معطيات التجمّع.

٢. **مقاطع الفهارس Index segments** : لتخزين معطيات الفهارس.

٣. **مقاطع التراجع Rollback segments** : يتم إنشاؤها من قبل مدير قاعدة المعطيات من أجل تخزين معلومات الإرجاع. وتستخدم هذه المعلومات بشكل مؤقت من أجل توليد معلومات قراءة متناسقة أو أثناء استرداد القاعدة أو للتراجع عن التحويلات غير المثبتة.

٤. **المقاطع المؤقتة Temporary segments** : يتم إنشاء هذه المقاطع عندما تحتاج تعليمات SQL إلى مساحة عمل مؤقتة من أجل إتمام تنفيذها. وتتم إعادة هذه المقاطع إلى النظام عند الانتهاء من التنفيذ.

الفضاءات الجدولية Tablespaces

تقسم قاعدة معطيات أوراكل إلى واحد أو أكثر من الفضاءات الجدولية. ويمكن لمدير قاعدة المعطيات استخدام هذه الفضاءات الجدولية في إجراء العمليات التالية :

- التحكم بمساحة تخزين المعطيات على القرص.

- تحديد نصيب مستخدمي القاعدة من مساحات التخزين.
- التحكم بإمكانية إتاحة المعطيات وذلك بتأهيل أو عدم تأهيل الفضاءات الجدولية.
- إجراء عمليات نسخ أو استرداد جزئية.
- حجز أماكن تخزين للمعطيات على عدة أقراص لتحسين الأداء.

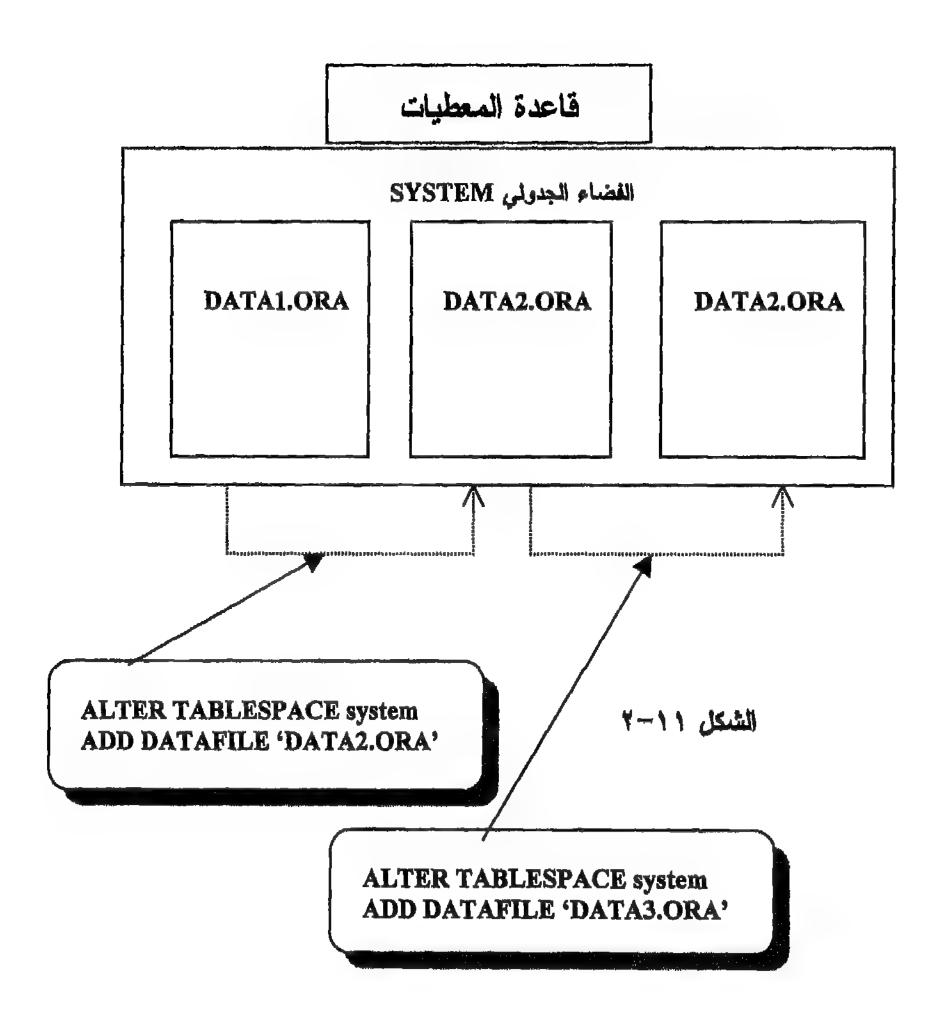
و كل قاعدة معطيات عند إنشائها تحتوي فضاءً جدولياً افتراضيــاً بالاســم System، يحتوي هذا الفضاء جداول قاموس المعطيات.

ويمكن أن تحتوي قاعدة معطيات صغيرة على الفضاء الجدولي SYSTEM فقــط، لكــن يفضل دائماً إنشاء فضاء جدولي آخر لتخزين معطيات المســتخدم بشــكل منفصــل عــن معلومات قاموس المعطيات، وهذا يعطيك مرونة أكثر في إجــراء عمليــات إدارة قواعــد المعطيات.

يمكن توسيع قاعدة المعطيات بإضافة ملف معطيات جديد أو أكثر وبالتالي زيادة مســـاحة التخزين الموافقة للفضاء الجدولي.

لنفترض مثلاً أن لدينا فضاءً جدولياً وحيداً SYSTEM يحوي على ملف معطيات وحيــد بالاســم DATA1.ORA يوضــح الشــكل ١١-٢ كيفيــة إضافــة ملفــي معطيــــات DATA3.ORA, DATA2.ORA. إلى هذا الفضاء الجدولي:

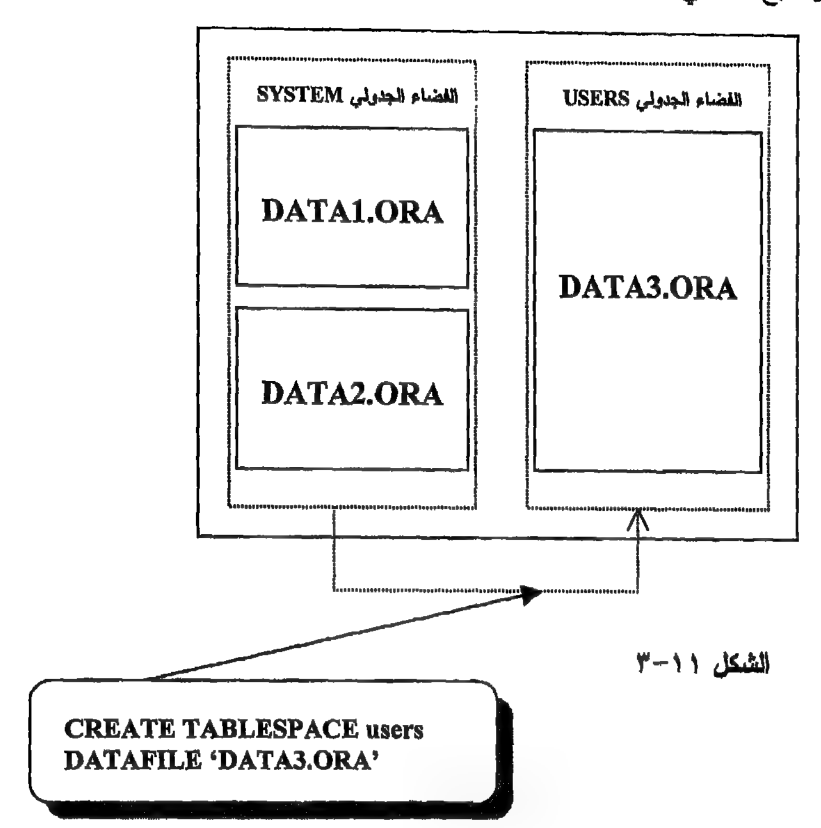




قاعدة المعطيات
الفضاء الجدولي SYSTEM
DATA1.ORA
DATA2.ORA
DATA2.ORA
ALTER TABLESPACE system
ADD DATAFILE 'DATA2.ORA'
ALTER TABLESPACE system
ADD DATAFILE 'DATA3.ORA'

الشكل ١١-٢

يمكن أيضاً لمدير قاعدة المعطيات DBA إنشاء فضاء جديد لزيادة حجم قاعدة المعطيات. يمكن توضيح ذلك في الشكل ١١-٣ :



الشكل ١١-٣

بالتالي فحجم الفضاء الجدولي هو مجموع أحجام ملفات المعطيات التي تكوّن هذا الفضاء، أما حجم قاعدة المعطيات فهو مجموع أحجام الفضاءات الجدولية التي تكوّن القاعدة.

ويمكن أيضاً لمدير قاعدة المعطيات جعل أي فضاء جدولي مؤهلاً Online أو غير مؤهل Offline وذلك عند فتح القاعدة، باستثناء وحيد فقط وهو أن الفضاء الجدولي SYSTEM يجب أن يبقى مؤهلاً دوماً بحيث يمكن لمستخدمي القاعدة الوصول إلى معطيات الفضاء.

يمكن لمدير النظام إلغاء تأهيل فضاء جدولي لأحد الأسباب التالية:

- لجعل جزء من قاعدة المعطيات غير متاح، وإتاحة الجزء المتبقي.
- لإجراء عملية نسخ احتياطي للفضاء الجدولي غير المؤهل (يمكن أيضاً إجراء عملية نسخ احتياطي لفضاء جدولي مؤهل وقيد الاستخدام).

- لجعل تطبيق ومجموعة الجداول المؤقتة الخاصة به غير متاحة أثناء عمليات تعديل وتصحيح هذا التطبيق.

في حال كون فضاء جدولي غير مؤهل، فإن أوراكل لا يسمح لأي من تعليمات SQL بالوصول إلى العناصر الموجودة في هذا الفضاء.

ولا يمكن جعل فضاء جدولي غير مؤهل إذا احتوى هذا الفضاء مقاطع تراجع فعالة.

أيضاً لايمكن إعادة تأهيل فضاء جدولي إلا من قبل قاعدة المعطيات التي ألغت تأهيله، والسبب هو أن معلومات قاموس المعطيات الضرورية للقيام بذلك موجودة ضمن الفضاء الجدولي SYSTEM المتعلق بهذه القاعدة فقط. ولا يمكن قراءة أو تحرير أي فضاء جدولي غير مؤهل إلا من قبل أدوات أوراكل. لذلك لايمكن نقل الفضاءات الجدولية من قاعدة معطيات إلى قاعدة معطيات أخرى.

وباستخدام عدة فضاءات جدولية لأنواع مختلفة من المعلومات، فإنه يمكن لمدير قاعدة المعطيات جعل بعض الفضاءات الجدولية غير مؤهلة للقيام ببعض الإجراءات، بينما تبقى بقية الفضاءات الجدولية مؤهلة ومتاحة للاستخدام.

لكن يجب الانتباه إلى حدوث بعض الإشكالات عند إلغاء تأهيل الفضاءات الجدولية، فمثلاً لنفترض أن لدينا فضاءين أحدهما للمعطيات والآخر للفهارس، ففي حال تم إلغاء تأهيل الفضاء الجدولي الذي يحتوي الفهارس، تبقى الاستعلامات قادرة على الوصول إلى معطيات الجداول لأنها قد لاتحتاج إلى فهرس لإجراء ذلك، بينما لا تستطيع الوصول إلى المعطيات في حال إلغاء تأهيل الفضاء الجدولي الذي يحتوي على جداول المعطيات.

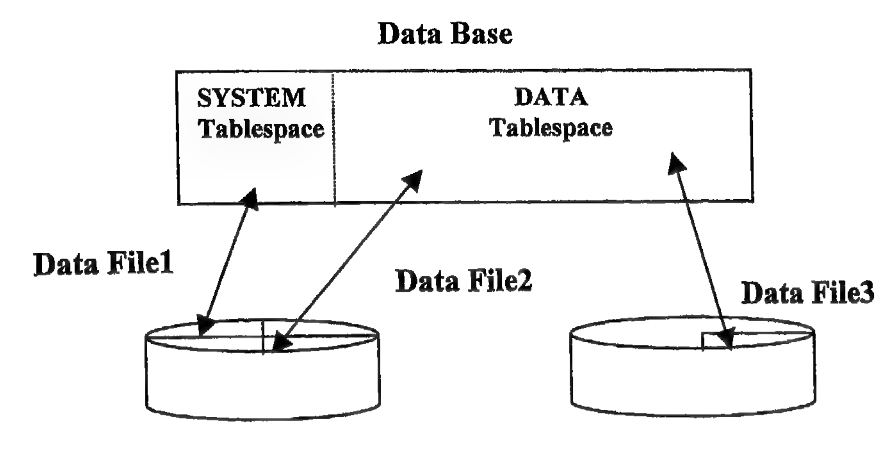
في المحصلة فإن أوراكل يختبر إن كانت هناك معلومات كافية في الفضاءات الجدولية المؤهلة لتنفيذ تعليمة فإنه يقوم بذلك، أما إذا احتاج إلى معطيات موجودة في فضاءات جدولية غير متاحة فإن التعليمة ستفشل.

يقوم أوراكل بحجز أماكن تخزين لكل أنواع المقاطع على شكل مدى، لذلك عندما يمتلئ المدى الموجود ضمن مقطع، يتم حجز مدىً جديد حسب الحاجة.

وعلى اعتبار أنه يمكن حجز مدىً جديد، لذلك يمكن أن يكون متجاوراً أو غير متجاور على القرص.

يتم تخزين المعطيات الخاصة بقاعدة معطيات أوراكل منطقياً في الفضاءات الجدولية وفيزيائياً في ملفات المعطيات المرتبطة مع الفضاء الجدولي الموافق.

ويوضح الشكل ١١-٤ علاقة الارتباط هذه :

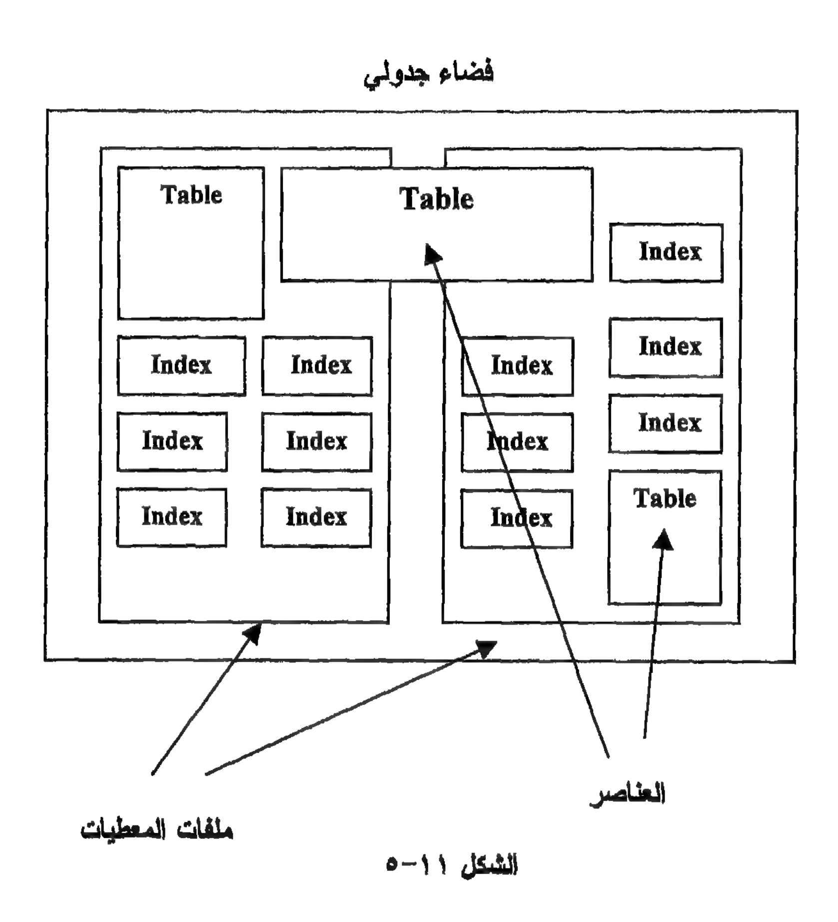


الشكل ١١-٤

وعلى الرغم من أن قواعد المعطيات والفضاءات الجدولية وملفات المعطيات والمقاطع مرتبطة بشكل وثيق إلا أنه توجد اختلافات بينها أهمها:

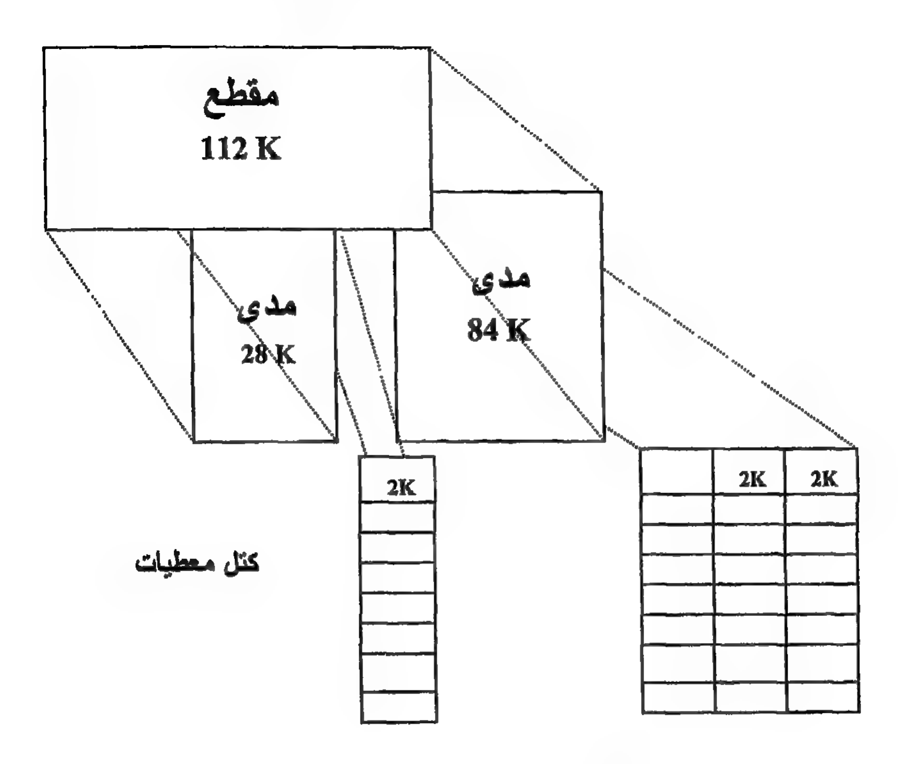
١. تتكون قاعدة معطيات أوراكل من فضاء أو أكثر من الفضاءات الجدولية والتي يتم تجميع معطيات القاعدة فيها.
٢. يتكون كل فضاء جدولي من ملف أو أكثر من ملفات المعطيات، بالتالي فإن ملفات معطيات الفضاء الجدولي تخزن المعطيات الموافقة فيزيائياً على قرص التخزين.
٣. تتكون قاعدة المعطيات من مجموعة من ملفات المعطيات، ويمكن كمثال أن نقوم بإنشاء قاعدة معطيات بسيطة تحتوي على فضاء جدولي وحيد يحتوي على ملف معطيات واحد. ويمكن إنشاء قاعدة معطيات أعقد قليلاً تحتوي على ثلاثة فضاءات جدولية يحتوي كل منها على ملفي معطيات.
٤. عندما يتم إنشاء عنصر كجدول أو فهرس ، فإنه يتم إنشاء مقطع موافق لهذا العنصر ضمن الفضاء الجدولي المرتبط.

يبين الشكل ١١-٥ الارتباط بين الفضاءات الجدوليـــة وملفــات المعطيــات والعنــاصر:

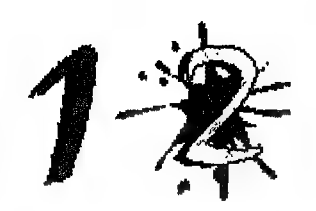


الشكل ١١-٥

أخيراً يبين الشكل ١١-٦ الارتباط بين المقاطع والمدى وكتل المعطيات:



الشكل ١١-٦





بنية ذاكرة أوراكل
Oracle Memory Structure

يستخدم أوراكل الذاكرة لتخزين المعلومات التالية:

- ☆ ترميز البرنامج الذي تمّ تنفيذه.
- ☆ معلومات عن الدورة المتصلة connected session حتى لو لم تكن فعالة.
- ☆ المعطيات المطلوبة خلال تنفيذ البرنامج.
- ☆ المعلومات المشتركة بين إجراءات أوراكل (معلومات القفل مثلاً).
- ☆ المعلومات المخبئية المخزّنة مؤقتاً في الذواكر الطرفية (ككتل المعطيات مثلاً).

تتضمن بنى الذاكرة الأساسية المرتبطة بأوراكل عدة مناطق هي:

- ⌘ مناطق ترميز البرمجيات Software Code Areas.
- ⌘ منطقة النظام العام System Global Area:
  - ذاكرة قاعدة المعطيات المخبئية Database Buffer Cache.
  - ذاكرة تسجيل الإرجاع Redo log Buffer.

- حوض المشاركة Shared Pool.

⌘ منطقة البرنامج العامة Program Global Area:

- منطقة التكديس Stack Areas.
- مناطق المعطيات Data Areas.

⌘ مناطق الفرز Sort Areas.

ويمكن لنظام أوراكل الاستفادة من الذواكر الافتراضية virtual memory في عمل البرمجيات وأجزاء أخرى من أوراكل، ويفضل إبقاء كامل SGA في الذاكرة الحقيقية.



مناطق ترميز البرمجيات Software Code Areas

عبارة عن أجزاء من الذاكرة تستخدم لتخزين الترميز الذي استخدم أو سيتم استخدامه.
ترميز أوراكل يخزن في منطقة البرمجيات في مكان مختلف عن برامج المستخدمين، ويكون حجم مناطق البرمجيات عادة ثابتاً ولا يمكن تغييره إلا بتعديل البرمجيات أو إعادة تثبيتها.
مناطق ترميز البرمجيات معدّة للقراءة فقط Read-Only ويمكن تثبيتها مشاركة Shared أو غير مشاركة Non-Shared.
ويمكن مشاركة ترميز أوراكل بحيث يستطيع جميع المستخدمين الوصول إليه دون الحاجة إلى امتلاك عدة نسخ منه في الذاكرة.

منطقة النظام العامة System Global Area (SGA)

عبارة عن مجموعة من بنى الذاكرة المشاركة التي يتم حجزها في كل عملية إقلاع لمخدّم أوراكل Oracle Server. وهي تحتوي على معطيات ومعلومات تحكم لهيئة قاعدة معطيات أوراكل واحدة.
وفي خال اتصال عدة مستخدمين بشكل متزامن بنفس الهيئة، تكون المعطيات في SGA الخاصة بالهيئة مشاركة بين المستخدمين.
تقسم المعلومات المخزنة في SGA إلى مناطق ذاكرة متعددة، تتضمن:

- **ذاكرة قاعدة المعطيات المخبئية المؤقتة Database Buffer Cache**: وتحتوي على معلومات قاعدة المعطيات، حيث تتضمن نسخ من كتل المعطيات تمّت قراءتها من ملفات المعطيات، هذه الذواكر المؤقتة مشاركة من قبل إجراءات المستخدم المتصلة بشكل متزامن مع الهيئة. يتم تحديد عدد الذواكر المؤقتة من خلال الوسيط DB-BLOCK-BUFFERS، أما حجم كل ذاكرة من هذه الذواكر فهو مساوٍ لحجم كتلة معطيات واحدة DB-BLOCK-SIZE.
- **ذواكر تسجيل الإرجاع المؤقتة Redo Log Buffer**: تحتوي على معلومات عن التغييرات التي تطرأ على قاعدة المعطيات. تخزن هذه المعلومات في مداخل الإرجاع

التي تحتوي على المعلومات اللازمة لإعادة بناء أو استرجاع التغييرات التي أجريت على القاعدة من خلال عمليات DROP,ALTER, CREATE, DELETE, UPDATE, INSERT. يمكن استخدام هذه المداخل لاسترداد قاعدة المعطيات عند الضرورة. تتم كتابة محتويات ذواكر الإرجاع المؤقتة على ملف الإرجاع الفعّال باستخدام إجراء الخلفية LGWR. ويمكن تحديد حجم ذاكرة الإرجاع المؤقتة من خلال الوسيط LOG-BUFFER.

- **حوض المشاركة Shared Pool**: عبارة عن منطقة في SGA تحتوي على بانيات Constructs مثل مناطق SQL المشاركة وذاكرة قاموس المعطيات المخبئية. يقوم أوراكل بتمثيل كل تعليمة SQL يقوم بتنفيذها بجزء مشترك وآخر خاص، وفي حال قيام مستخدمان مختلفان بتنفيذ نفس تعليمة SQL فإنه يعيد استخدام نفس الجزء المشترك لهما.

منطقة SQL المشتركة Shared SQL Area هي منطقة ذاكرة تحتوي على شجرة عبور Parse tree ومخطط تنفيذ execution plan لتعليمة SQL وحيدة. أما منطقة SQL الخاصة فهي منطقة ذاكرة تحتوي على معطيات كالمعلومات المرتبطة وذواكر التنفيذ المؤقتة.

أما ذاكرة قاموس المعطيات المخبئية Data Dictionary Cache فهي عبارة عن مجموعة من الجداول والمشاهد التي تحتوي على مراجع للمعلومات عن قاعدة المعطيات كأسماء جميع الجداول والمشاهد في القاعدة، كذلك أسماء وأنماط الأعمدة في الجداول وسماحيات جميع المستخدمين.

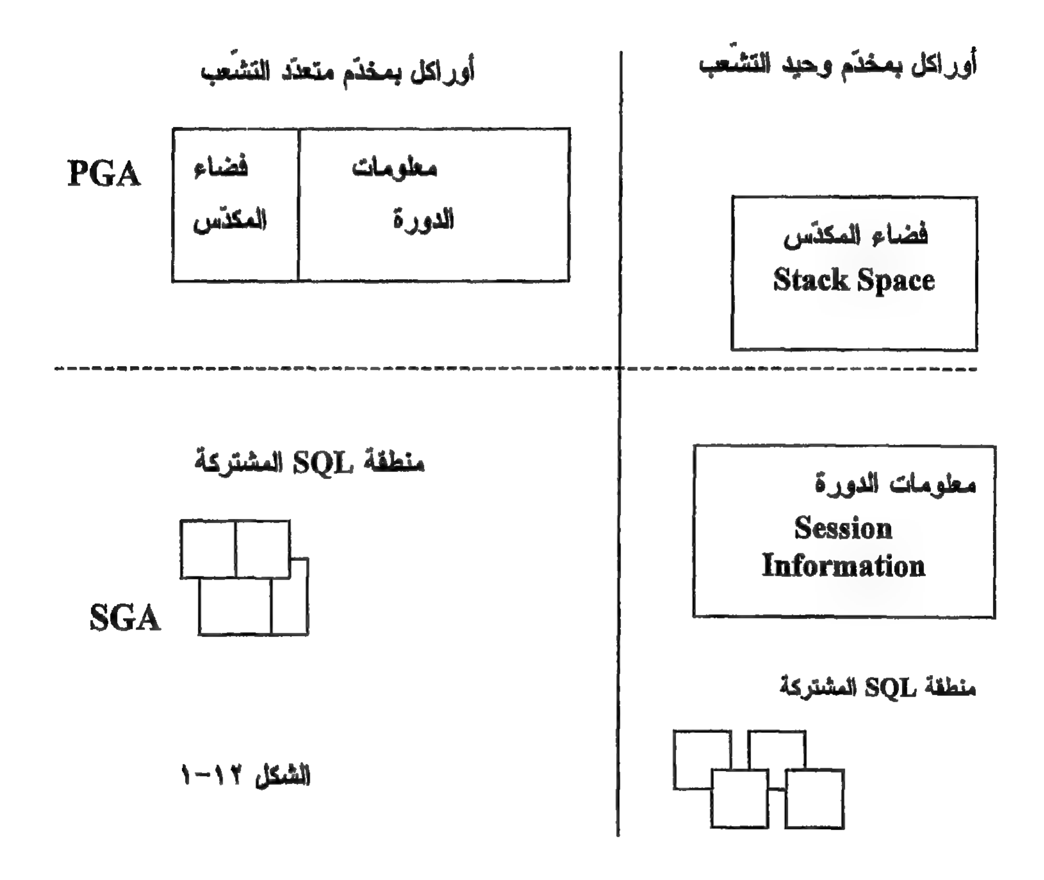
ويمكن تحديد حجم حوض المشاركة من خلال الوسيط SHARED-POOL-SIZE.

منطقة البرنامج العامة (PGA) Program Global Area

عبارة عن منطقة ذاكرة تحتوي على معطيات ومعلومات تحكم لإجراء وحيد (إجراء مخدّم Server Process أو إجراء خلفية Background Process).

يــتم حجــز PGA عــندما يقوم إجراء مستخدم بالاتصال مع قاعدة معطيات أوراكل وإنشاء دورة Session.

لذلــك فــإن محتوى PGA يتغير بالاعتماد على عمل الممثل المرتبط به أو عدمه في مخدّم متعدد التشعّب multi-threaded server (انظر الشكل ١٢-١).



الشكل ١٢-١

يحتوي PGA دائماً على فضاء مكدس Stack Space وهو عبارة عن ذاكرة يتم حجزها لاحتواء متحولات ومصفوفات الدورات ومعلومات أخرى.

ويحتاج PGA دوماً إلى هيئة لا تعمل في مخدّم متعدد التشعّب، وإلى ذاكرة إضافية لدورة المســتخدم كمــنطقة SQL الخاصة ومعلومات أخرى. ويتعلق حجم PGA بنظام التشغيل وهو غير ديناميكي. وعندما يكون المخدّم والزبون على جهازين مختلفين، يتم حجز PGA

في مخدّم قاعدة المعطيات أثناء الاتصال به، وعندما لا توجد ذاكرة كافية للاتصال، يحدث الخطأ.

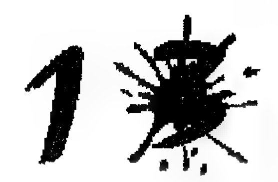
وتؤثر الوسطاء التالية على حجم PGA:

OPEN-LINKS
DB-FILES
LOG-FILES

مناطق الفرز Sort Area

وهـي عـبارة عن مناطق ذاكرة لفرز المعطيات ويتم إنشاؤها عند طلب إجراء المستخدم لعمـلية فرز. ويمكن أن تتزايد هذه المناطق حسب كمية المعطيات التي سيتم فرزها لكنها تبقى محددة دوماً بالوسيط SORT-AREA-SIZE.







بنية إجراءات أوراكل
Oracle Process Structure

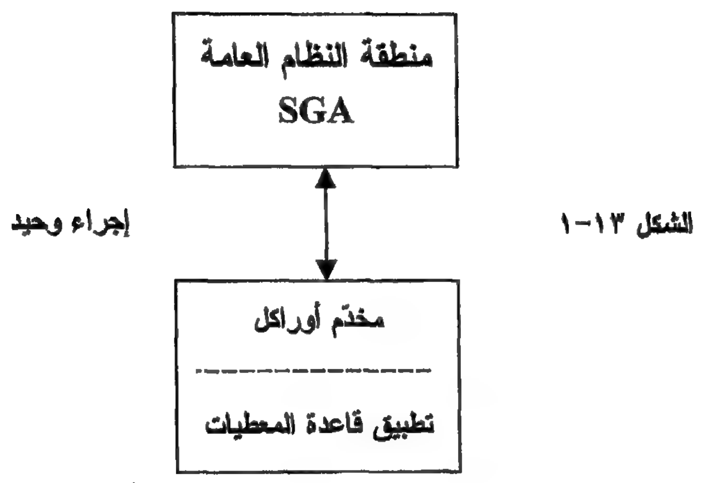
الإجراء هو ميكانيكية ضمن نظام التشغيل يمكنها تنفيذ سلسلة مـــن الخطـــوات. عادةً يكون لكل إجراء منطقة ذاكرة خاصة به يستطيع العمل ضمنها.

بنية الإجراء في نظام كنظام أوراكل هامة جداً لأنها تعـــرّف إمكانيـــة إجراء وتنفيذ المهام المتعددة التي يمكن أن يقوم بها النظام، لذلك تـــمّ تصميـــم إجـــراءات أوراكل لإعطاء الأداء الأفضل. سنقوم فيمايلي بتوضيح هذه البنية في حالتين: هيئة أوراكل بإجراء وحيد وهيئة أوراكل متعددة الإجراءات.

هيئـــة أوراكـــل بـــإجراء وحيـــد Single Process ORACLE Instance

وهي حالة المستخدم الوحيد لنظام أوراكل حيث يتم تنفيذ جميع ترميزات أوراكل من قبـــل إجراء واحد.

يوضح الشكل ١٣-١ بنية الإجراء الوحيد:



الشكل ١٣-١ إجراء وحيد

في بيئة الإجراء الوحيد، يمكن لمستخدم واحد فقط الوصول إلى هيئة أوراكل، ولا يمكـــن لعدة مستخدمين الوصول إلى قاعدة المعطيات بشكل متزامن.

حالة الإجراء الوحيد يمكن أن نراها في أنظمة أوراكـــل التـــي تعمـــل علـــى الحواســـيب الشخصية كالتي تعمل تحت نظام MS-DOS أو WINDOWS95.

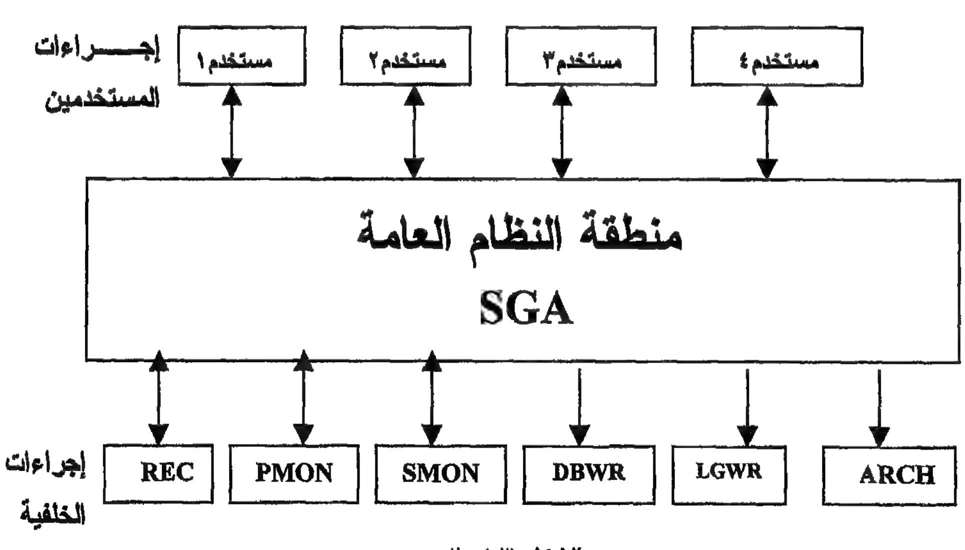
هيئة أوراكل متعددة الإجـــــراءات Multiple-Process ORACLE Instance

وهي حالة نظام أوراكل متعدد المستخدمين، حيث يقوم باستخدام عدة إجراءات لتنفيذ أجزاء مختلفة ضمن أوراكل واستخدام إجراءات منفصلة لكل مستخدم متصل بالنظام.

لكل إجراء عمل خاص، وبتقسيم عمل نظام أوراكل وتطبيقات قاعدة المعطيات إلـــى عـــدة إجراءات، يمكن لعدة مستخدمين وعدة تطبيقات الاتصال بشكل متزامن مع هيئـــة قـــاعدة معطيات وحيدة.

أغلب أنظمة قواعد المعطيات هي أنظمة متعددة المستخدمين لأن أحد أهم مـــيّزات قواعـــد المعطيات هي إدارة المعطيات المطلوبة من قبل عدة مستخدمين في الوقت نفسه.

يوضح الشكل ١٣-٢ حالة هيئة أوراكل بعدة إجراءات:



الشكل ١٣-٢

نلاحظ من الشكل السابق أن لكل مستخدم متصل بالقاعدة إجـــراء مســـتخدم منفصـــل وأن العديد من إجراءات الخلفية back ground processes تستخدم لتنفيذ أوراكل.

يمكن تقسيم الإجراءات إلى مجموعتين: إجراءات المستخدم وإجراءات أوراكل.

إجراءات المستخدم User processes

عندما يقوم المستخدم بتنفيذ برنامج تطبيق ما كبرنامج Pro*C أو أحد أدوات أوراكل مثل Server Manager، يتم توليد إجراء مستخدم لتنفيذ هذا البرنامج. تقوم هذه الإجراءات أيضاً بتمرير تعليمات SQL إلى إجراء مخدّم وتلقي النتيجة.

إجراءات أوراكل ORACLE processes

في الأنظمة متعددة الإجراءات يتم التحكم بأوراكل من قبل نوعين من الإجراءات: إجراءات المخدّم وإجراءات الخلفية.

إجراءات المخدّم Server processes

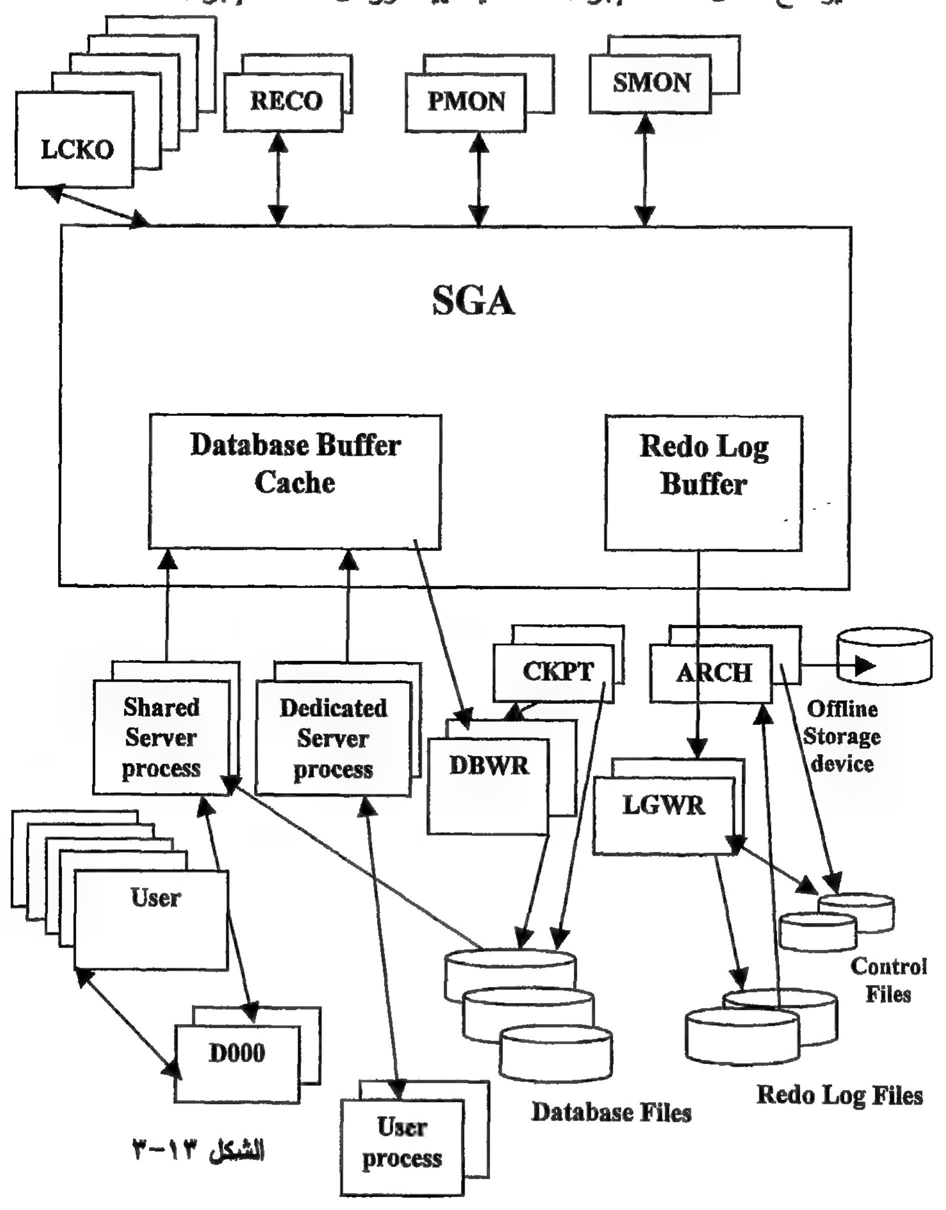
تستخدم لمعالجة طلبات إجراءات المستخدم المتصل بهيئة أوراكل. ويمكن أن يتم إنشاء هذه الإجراءات لإنجاز العمليات التالية:

- عبور parse وتنفيذ تعليمات SQL الصادرة عن التطبيقات.
- قراءة كتل المعطيات وتخزينها في الذواكر المشتركة لقاعدة المعطيات Shared database buffers الموجودة في ذاكرة SGA في حال عدم وجود هذه الكتل فيها من قبل.
- إرجاع نتائج يمكن للتطبيق معالجتها.

إجراءات الخلفية background processes

يمكن لهيئة أوراكل أن تمتلك عدداً كبيراً من إجراءات الخلفية.

يوضح الشكل ١٣-٣ إجراءات الخلفية لهيئة أوراكل متعددة الإجراءات:



الشكل ١٣-٣

يمكن أن نحدد إجراءات الخلفية على الشكل:

- **Database Writter (DBWR):** تتم جميع عمليات كتابة محتويات الذواكـــر المؤقتة في ملفات المعطيات باستخدام هذا الإجراء. ويقوم إجـــراء DBWR بـــإدارة الذواكر المؤقتة بحيث يمكن لإجراءات المستخدم إيجاد ذواكر مؤقتة فارغة بشكل دائم.

- **Log Writer (LGWR):** يقوم هذا الإجراء بكتابة محتويات ذواكر الإرجـــاع المؤقتة ضمن ملفات الإرجاع.

- **Check point (CKPT):** عند حدوث نقطة التدقيق يتم تعديل ترويسات جميـــع ملفات المعطيات للدلالة على نقطة التدقيق. يتم تنفيذ هذا العمل عادةً من قبل الإجـــراء LGWR. ويمكن تفعيل إجراء CKPT في حال عدم وجود ملفات معطيـــات كافيـــة وذلك من أجل فصل عمل معالجة نقطة التدقيق عن بقية الأعمـــال التـــي يقـــوم بـــها الإجراء LGWR.

في أغلب التطبيقات فإن إجراء CKPT غير ضروري، لأنه إذا كانت قـــاعدة المعطيـــات تمتلك عدداً كبيراً من ملفات المعطيات وقلّ أداء إجراء LGWR بشكل واضح أثناء نقـــاط التدقيق، نحتاج عندها إلى تفعيل إجراء CKPT ويمكن القيام بذلك من خلال تحديد قيمـــة الوسيط CHECKPOINT-PROCESS.

- **System Monitor (SMON):** يقوم هذا الإجراء بعملية استرداد هيئة أوراكل أثناء إقلاعها، كذلك فإن هذا الإجراء مسؤول عن تفريغ المقاطع المؤقتة التي لم تعـــد قابلة للاستخدام كما أنه مسؤول عن تجميع الكتل المدى الفارغة المنفصلة لإنشاء كتـــل فارغة متصلة وكبيرة.

- **Process Monitor (PMON):** هذا الإجراء مسؤول عـــن عمليـــة اســـترداد إجراء في حال فشل إجراء مستخدم. كذلك هو مسؤول عن مسح الذاكـــرة المخبئيـــة

وتحرير المصادر التي كان يستخدمها الإجراء. يقوم الإجراء PMON بشكل دوري بالتحقق من حالة الموزّع dispatcher وإجراءات المخدّم server processes، وإعادة تشغيل أي من الإجراءات التي قتلت Killed (عدا تلك التي قام أوراكل بقتلها من تلقاء نفسه).

- **RECOVERER (RECO):** يستخدم هذا الإجراء مع الخيار الموزّع distributed option لحل المشاكل التي تحدث أثناء التحويلات الموزعة distributed transactions بشكل تلقائي.

- **Archiver (ARCH):** يقوم هذا الإجراء بنسخ ملفات الإرجاع الفعالة ضمن سواقة التخزين المحددة عندما تمتلئ هذه الملفات. ويكون هذا الإجراء موجود فقط في حال استخدمت ملفات الإرجاع بنمط ARCHIVELOG وكانت الأرشفة التلقائية فعالة.

- **LOCKn(LCK):** يتم استخدامه مع خيار المخدّم المتوازي Parallel Server ويمكن استخدام عشرة إجراءات للقفل (LCK0,.....,LCK9) من أجل عمليات قفل الهيئة الداخلية inter-instance locking. ومع ذلك فإن إجراء LCK واحد (LCK0) كافٍ لأغلب أنظمة المخدّمات المتوازية.

- **Dispatcher processes (Dnnn):** تسمح لإجرائيات المستخدم user processes بالمشاركة في عدد محدد من إجراءات المخدّم server processes. وبدون الموزّع يحتاج كل إجراء مستخدم إلى إجراء مخدّم مكدس واحد dedicated server process.

بينما في المخدّمات متعددة التشعبات multi-threaded server، فإن عدداً قليلاً من إجراءات المخدّم المشاركة يكفي لنفس عدد المستخدمين. من أجل ذلك فإنه في الأنظمة التي يتعامل معها عدد كبير من المستخدمين، يمكن للمخدّمات متعددة التشعبات دعم هذا

العدد الكبير من المستخدمين خاصة في بيئات المخدّم/الزبون والتي يعمـــل فيــها تطبيــق الزبون في جهاز مختلف عن المخدّم.

يمكن إنشاء عدة إجراءات موزّع إلى هيئة قاعدة معطيات وحيدة، ويجب إنشاء موزّع واحد على الأقل لكل بروتوكول شبكة يمكّن المستخدمين من الاتصال بأوراكل. أخيراً يجب على مدير قاعدة المعطيات تشغيل عدداً أمثل من إجراءات الموزّع اعتماداً على محددات نظــام التشغيل المتعلقة بعدد الاتصالات لكل إجراء، ويمكنه إضافة أو حذف إجـــراءات مــوزّع dispatcher processes خلال عمل الهيئة.







بنى إضافية
Additional Structures

يوجد العديد من المفاهيم والبنى الإضافية المساعدة أثناء عمل قاعدة المعطيات، كملفات الأثر Trace Files، وملفات الإنذار Alert Files. سنقوم في هذا الفصل بشرح عمل هذه الملفات إضافة إلى بعض المفاهيم الأساسية المستخدمة كمحدد أوراكل Oracle SID، والتحويلات Transactions، وهيئة أوراكل Oracle Instance.

ملفات الأثر Trace Files

يمكن لكل مخدم وكل إجراء خلفية الكتابة على ملف أثر موافق. فعندما يكشف إجراء عن خطأ داخلي فإنّه يقوم بكتابة المعلومات المتعلقة بهذا الخطأ في ملف الأثر الخاص به.

يحدد الوسيط BACKGROUND_DUMP_DEST موقع ملفات الأثر في حال تمّت كتابة معلومات هذه الملفات من قبل إجراءات الخلفية، أما الوسيط BACKGROUND_DUMP_USER فيحدد موقع ملفات الأثر في حال تمّت كتابة معلومات هذه الملفات من قبل إجراءات المخدم.

ويجب تحديد الوسيط SQL-TRACE بالقيمة TRUE حتى تستطيع ملفات الأثر الكتابة عن سلوك إجراءات المخدم. وتستطيع كل دورة تأهيل أو عدم تأهيل الأثر باستخدام تعليمة ALTER SESSION مع الوسيط SQL-TRACE، فمثلاً تسمح التعليمة التالية بتفعيل الكتابة على ملف أثر للدورة:

```
ALTER SESSION SET SQL-TRACE=TRUE;
```

ملفات الإنذار Alert Files

لكل قاعدة معطيات ملف إنذار يحتوي على سجلات متزامنة للرسائل والأخطاء، تتضمن:

* كل الأخطاء الداخلية (ORA-600) وأخطاء مقاطعة الكتل (ORA-1578) وأخطاء الإخفاقات (ORA-60) التي تحدث.
* عمليات الإدارة.
* العديد من الرسائل والأخطاء المرتبطة بعمل إجراءات المخدم المشتركة وإجراءات المرسل.
* الأخطاء التي تحدث أثناء التحديث التلقائي للقطة قاعدة المعطيات.

يتم تحديد موقع ملف الإنذار من خلال الوسيط BACKGROUND_DUMP_DEST.

هيئة أوراكل Oracle Instance

أي قاعدة معطيات أوراكل فعالة ترتبط مع هيئة أوراكل. وعند تشغيل قاعدة معطيات على مخدم قاعدة المعطيات يتم حجز مساحة ذاكرة هي مساحة النظام العامة (SGA) System Global Area ويتم تشغيل إجراء أو أكثر من إجراءات أوراكل.

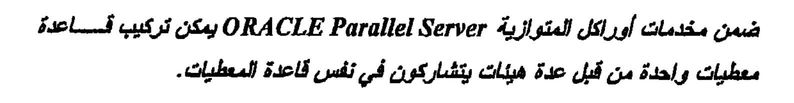
إذاً هيئة قاعدة معطيات أوراكل عبارة عن ذاكرة SGA مع إجراءات أوراكل Oracle Processes.

تقوم الذاكرة والإجراءات الخاصة بهيئة أوراكل بالعمل بشكل فعال لإدارة معطيات القاعدة وتخدم مستخدمي هذه القاعدة.

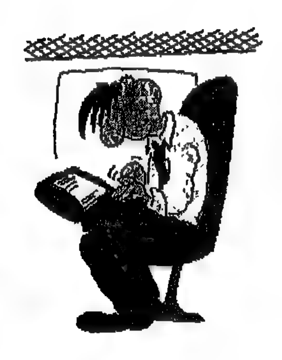
عند إقلاع قاعدة معطيات، يتم أولاً تشغيل هيئة القاعدة ومن ثم تركيب القاعدة mount من قبل هذه الهيئة.

ويمكن لعدة هيئات العمل على نفس الجهاز بشكل متزامن، ولكل هيئة قاعدتها الفيزيائية الخاصة بها.

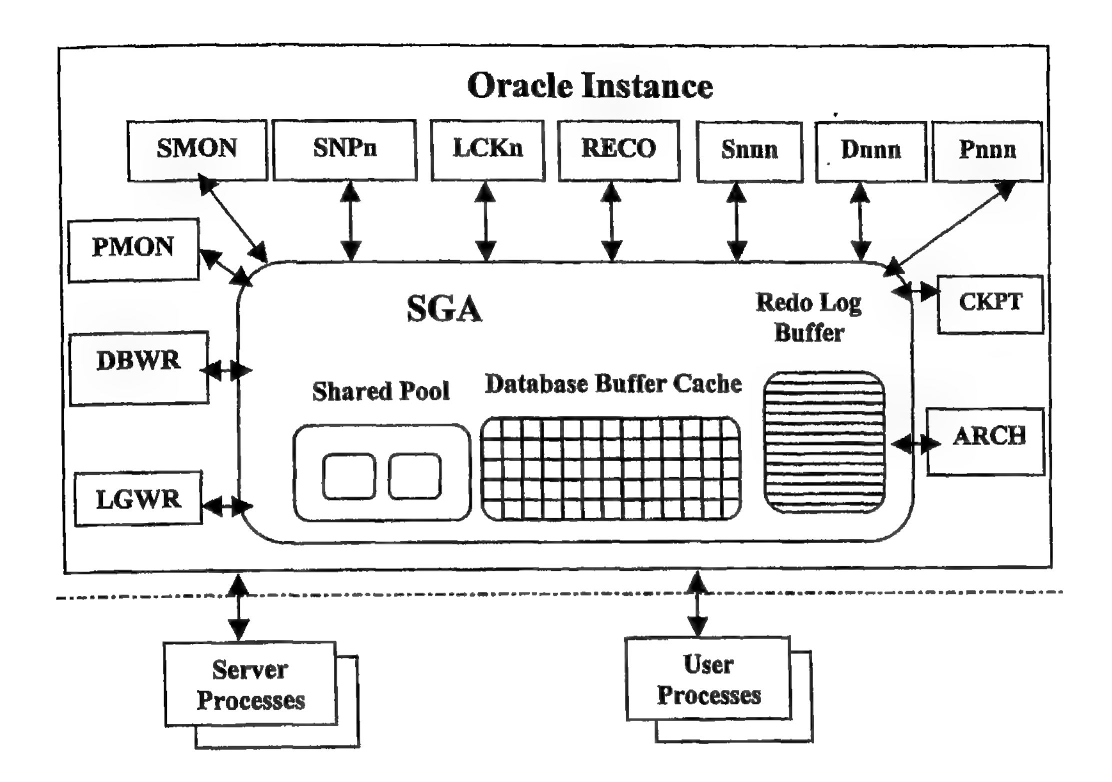




ضمن مخدمات أوراكل المتوازية ORACLE Parallel Server يمكن تركيب قاعدة معطيات واحدة من قبل عدة هيئات يتشاركون في نفس قاعدة المعطيات.



يوضح الشكل ١٤-١ بنية هيئة أوراكل :



الشكل ١٤-١



التحويلات *Transactions*

التحويل عبارة عن وحدة عمل منطقية تتكون من تعليمة أو أكثر من تعليمات *SQL* ويمكن لجميع تعليمات *SQL* المكونة للتحويل أن تثبت في قاعدة المعطيات *committed* أو تلغى من قاعدة المعطيات *rolled back* . يبدأ التحويل من أول تعليمة *SQL* وينتهي عند طلب *commit* أو *rollback*.



يبين الشكل ١٤-٢ تحويل لنقل مبلغ من حساب إلى حساب آخر ضمن قاعدة معطيات.

بداية تحويل

```
Decrement SavingsAccount
UPDATE savings-accounts
  SET balance= balance-50
WHERE account=3209;
```

```
Incerment checking Account
UPDATE checking-accounts
  SET balance=balance+50
WHERE account=3208
```

```
Record in transaction Journal
INSERT INTO journal VALUES
(journal-seq-NEXT VAL,'1B',
3209,3208,500);
```

```
End Transaction
COMMIT WORK;
```

الشكل ١٤-٢

نهاية تحويل

محدّد النظام SID

يتم التعرّف على هيئة أوراكل بمحدد النظام SID (System identifier)، وهـو وحـده يتعرف على هذه الهيئة، ويتم استخدامه من قبل أدوات أوراكل وأجزاء الشبكة مـن أجـل الاتصال مع هيئة أوراكل المحدّدة.

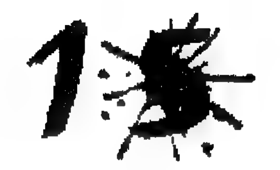
يمكن أن يصل طول محدد أوراكل إلى أربعة أحرف فقط، ويتم تحديد قيمته مـن خـلال متحول الوسط ORACLE_SID.

يمكنك من خلال سطر أوامر نظام WINDOWS NT تحديد قيمـة SID باسـتخدام التعليمة:

Set ORACL_SID = ORCL

ويتم تثبيت قيمة محدد النظام في سجل النظام Registry.







قاموس المعطيات
Data Dictionary

كل عنصر وكل فضاء جدولي يضاف إلى قاعدة المعطيات، يدخل له توصيف في قاموس المعطيات. قاموس المعطيات هو مجموعة من الجداول المعدّة للقراءة فقط والتي تخزّن وتوثق معلومات عن كل العناصر الموجودة في هذه القاعدة. هذه الجداول هي ملك لمدير النظام وتنشأ بنشوء قاعدة المعطيات، ولايستطيع أي كان الوصول إلى هذه الجداول.

يمكن الحصول مثلاً على المعلومات التالية من قاموس المعطيات:

- أسماء مستخدمي أوراكل.
- سماحيات المستخدمين.
- أسماء مخططات العناصر...إلخ.

ويتم وضع جميع محتويات قاموس المعطيات في حساب بالاسم SYS. حيث لا يمكن للمستخدمين العاديين إجراء أي تعديل ضمن هذا الحساب.

يمكـن للمستخدمين العاديين الوصول إلى قاموس المعطيات من خلال مشاهد المعاينة التي يتبعها النظام وهي:

- مشاهد معاينة المستخدم (ماالذي يحتويه مخطط المستخدم) USER.
- مشهد معاينة مستخدم موسع (ماالذي يمكن للمستخدم الوصول إليه) ALL.
- مشـهد معايـنة مدير قاعدة المعطيات (ماالذي يمكن لجميع المستخدمين الوصول إليه) DBA.

أين يتم تخزين قاموس المعطيات؟

عـندما يتم إنشاء قاعدة معطيات جديدة ينشأ معها فضاء جدولي يدعى SYSTEM وينشأ مخطـط عناصر يدعى SYSTEM يحتوي نفس العناصر الموجودة في الفضاء الجدولي. في هذا الفضاء الجدولي وملفاته يخزّن النظام قاموس المعطيات.

كيف السبيل للدخول إلى قاموس المعطيات؟

عـند إنشـاء قـاعدة معطيـات ينشـأ معها حسابان هما SYSTEM/MANAGER و SYS/CHANGE_ON_INSTALL. الأول لمديـر النظام ويمتلك جميع حقوق مدير الـنظام، والـثاني لاستعراض قاموس المعطيات. عند الدخول باسم الحساب الثاني يمكننا اسـتعراض القـاموس، ولكن غير مسموح لنا الكتابة فيه فالقاموس هو للقراءة فقط ووحده نظام أوراكل من يستطيع التعديل فيه.

كيف يتم تعديل قاموس المعطيات؟

عـند كل عملية إنشاء أو تعديل أو حذف عنصر من العناصر أو فضاء جدولي أو مخطط عناصـر يقـوم أوراكـل بإجراء التعديلات المناسبة في قاموس المعطيات ليعكس الوضع الحالي للقاعدة.

مم يتألف قاموس المعطيات؟

يتألف قاموس المعطيات من جداول أساسية تخزن فيها المعلومات بطريقة وشيفرة خاصة بالنظام يصعب على المستثمر العادي متابعتها وفهمها. كما يحتوي على مشاهد إلى لتنسيق المعلومات بشكل مفهوم للمستخدم، وبعض هذه المشاهد متاح للاستخدام العام وذلك لاطلاع المستثمر العادي عليها.

لنجرب المثال التالي على مستوى حساب SCOTT :

```
SELECT * FROM all_tables;
```

إن all_tables هو عبارة عن مشهد منسق ومرتب ليستعرض كل الجداول الموجودة بقاعدة المعطيات.

لنجرب الاستعلام التالي :

```
SELECT *from all_objects;
```

حيث أن all_objects أيضاً هو مشهد من مشاهد قاموس المعطيات ممنوح للحساب PUBLIC.

ما هي الجداول الأساسية لقاموس المعطيات؟

تحتوي الجداول والتجمعات التالية على تعاريف جميع العناصر التي يقوم المستخدمون بإنشائها في قاعدة المعطيات :



اسم الجدول أو التجمع	المحتوى
SEG$	جميع المقاطع المعرفة في قاعدة المعطيات (مع المقاطع المؤقتة).
OBJ$	جميع عناصر المستخدم المعرفة في القاعدة.
UNDO$	مقاطع التراجع المعرفة في القاعدة.
FET$	المدى الحر وغير المحجوز في المقاطع.
UET$	المدى المحجوز في المقاطع.
TS$	الفضاءات الجدولية المعرفة في القاعدة.
FILE$	الملفات التي تكوّن القاعدة.

TAB$	الجداول المعرفة في القاعدة.
CLU$	التجمعات المعرّفة في القاعدة.
IND$	الفهارس المعرّفة في القاعدة.
ICOL$	الأعمدة التي تحتوي على فهارس معرّفة عليها.
COL$	الأعمدة المعرّفة في الجداول.
CON$	القيود المعرفة في القاعدة.
CDEF$	تعريفات القيود في CON$.
CCOL$	الأعمدة التي تحتوي على قيود معرّفة عليها.
USER$	المستخدمين USERS و ROLES المعرفة في القاعدة.
TSQ$	نصيب الفضاءات الجدولية للمستخدمين.
C-OBJ	التجمعات التي تحتوي على:COL$,IND$,ICOL$,CLU$,TAB$.
C-TS	التجمعات التي تحتوي على: FILE$, TS4, FET$.
C-FILE#-BLOCK#	التجمعات التي تحتوي على: UET$, SEG$.
C-USER#	التجمعات التي تحتوي على : TSQ$, USER$.
C-COBJ#	التجمعات التي تحتوي على :CCOL$, CDEF$.

إظهار معلومات مخطط العناصر Schema Objects

يزودنا قاموس المعطيات بمجموعة من المشاهد التي تعطينا معلومات عن مخطط العناصر. تلخص القائمة التالية المشاهد المتعلقة بمخطط العناصر (سنستعرض في الفصول القائمة جميع هذه المشاهد بشكل مفصّل):

- *ALL_OBJECTS, USER_OBJECTS, DBA_OBJECTS*
- *ALL_CATALOG, USER_CATALOG, DBA_CATALOG*
- *ALL_TABLES, USER_TABLES, DBA_TABLES*
- *ALL_TAB_COLUMNS, USER_TAB_COLUMNS, DBA_TAB_COLUMNS*
- *ALL_TAB_COMMENTS, USER_TAB_COMMENTS*

- ALL_COL_COMMENTS, USER_COL_COMMENTS, DBA_COL_COMMENTS
- ALL_VIEWS, USER_VIEWS, DBA_VIWS
- ALL_INDEXES, USER_INDEXES, DBA_INDEXES
- ALL_IND_COLUMNS, USER_IND_COLUMNS, DBA_IND_COLUMNS
- USER_CLUSTERS, DBA_CLUSTERS
- USER_CLU_COLUMNS, DBA_CLU_COLUMNS
- ALL_SEQUENCES, USER_SEQUENCES, DBA_SECUENCES
- ALL_DEPENDENCIES, USER_DEPENDENCIES, DBA_DEPENDENCIES

وتحتوي المشاهد التالية على معلومات عن مقاطع قاعدة المعطيات :

- USER_SEGMENTS, DBA_SEGMENTS

أما المشاهد التالية فتحتوي على معلومات عن المدى في قاعدة المعطيات :

- USER_EXTENTS, DBA_EXTENTS
- USER_FREE_SPACE, DBA_FREE_SPACE



إظهار مخطط العناصر بحسب النمط :

```
SELECT object_name, object_type, FROM user_objects;
```



إظهار معلومات الأعمدة :

```
SELECT table_name,column_name,data_default
FROM user_tab_columns
WHERE table_name ='DEPT' OR table_name ='emp';
```



إظهار معلومات عامة عن المقاطع:

```
SELECT segments_name,
tablespace_name,bytes,blocks,extents
FROM user_tab_columns
WHERE segment_type = 'ROLLBACK';
```



إظهار معلومات عامة عن المدى:

```
SELECT segment_name,bytes,blocks
FROM sys.dba_extents
WHERE segment_type = 'ROLLBACK';
```



إظهار معلومات عن الأماكن الفارغة في قاعدة المعطيات :

tablespace_name,file_id,bytes,blocks
FROM sys.dba_free_space;

إنشـــاء قـــاموس المعطيـــات Creating The Data Dictionary

عندما تقوم بإنشاء قاعدة معطيات، يتم وبشكل تلقائي توليد قاموس المعطيات. لذلك فإنه في أي وقـــت تكـــون فيه قاعدة المعطيات فعالة، يقوم أوراكل بتعديل قاموس المعطيات وذلك كاســـتجابة لكـــل تعليمة من تعليمات لغة تعريف المعطيات DDL (Data Definition Language).

يجـــب أن يتم أولاً إنشاء جداول قاموس المعطيات وذلك من أجل تخزين جميع المعلومات عن العناصر التي يقوم المستخدم بتعريفها وإنشائها.

ويـــتم تحديد قيمة الوسيط INIT-SQL-FILES من أجل معرفة أسماء ملفات السرد التي ســـيتم تـــنفيذها تـــلقائياً بعد إنشاء قاعدة المعطيات وتختلف أسماء الملفات الافتراضية لهذا الوسيط حسب نظام التشغيل.



مثلاً يمكنك إضافة أسماء لملفاتك بعد أسماء الملفات الافتراضية:

INIT_SQL_FILES= (CATALOG. SQL, CATPROC. SQL, ACME_DBA. SQL)

في هذا المثال ACME_DBA.SQL هو ملف إضافي يتم تشغيله أثناء إنشاء قاعدة المعطيات.

بشكل عام، تقوم ملفات SQL الابتدائية هذه بالمهام التالية:

١- تعريف الفضاء الجدولي SYSTEM و مقطع التراجع SYSTEM.

٢- تعريف جداول قاموس المعطيات في الفضاء الجدولي SYSTEM.

٣- شحن المعطيات لبعض جداول قاموس المعطيات.

يوضـــح المـــلحق١ ملفات السرد المطلوبة لمخدم أوراكل مع الخيارات الموافقة، وهي تنفذ تلقائياً عندما تقوم بإنشاء قاعدة المعطيات.


الباب الرابع



١٦ . أدوات إدارة أوراكل.





أدوات إدارة أوراكل

Oracle Administration Tools

سنقوم فـــي هذا الفصل بالتعرف على بعض عناصر أوراكل المطلوب تحديدها من قبل مدير قاعدة المعطيات DBA مثل حسابات مدير القاعدة DBA ووظـــائف مدير القاعدة DBA Roles. بالإضافة إلى بعض المهام التي يجب على مدير القاعدة إنجازها وتتضمن:

- تركيب واستخدام برنامج Enterprise Manager.
- استخدام برنامج Server Manager.

حسابات مدير قاعدة المعطيات The DBA Accounts

حتى يتمكن مدير قاعدة معطيات أوراكل، يجب أن يمتلك بعض الامتيازات الخاصة والتي تسمح له بتنفيذ الأوامر التي لا يمكن لبقية المستخدمين إنجازها. بعد تركيب نظام أوراكل، فإنه يتم إنشاء عدة حسابات مع امتيازات خاصة، هذه الحسابات هي:

١- **INTERNAL**: يتم إنشاء هذا الحساب بشكل رئيسي من أجل التوافق مع النسخ السابقة من أوراكل، كما أنه الحساب المسؤول عن إجراء عمليات إقلاع وإطفاء ممثل أوراكل. ويمكن استخدام هذا الحساب حتى لو لم يكن قد تم إنشاء قاعدة المعطيات أو تم إقلاع هيئة أوراكل.

٢- **SYS**: يتم إنشاء هذا الحساب تلقائياً عند إنشاء قاعدة المعطيات. ويستخدم هذا الحساب بشكل رئيسي لإدارة قاموس المعطيات.

يمنح هذا الحساب امتياز DBA إضافة إلى الوظيفتين: RESOURCE, CONNECT.

٣-**SYSTEM**: يتم أيضاً إنشاء هذا الحساب تلقائياً عند إنشاء قاعدة معطيات. ويستخدم بشكل رئيسي لإنشاء الجداول والمشاهد الضرورية لعمليات إدارة قواعد المعطيات ويمنح هذا الحساب الامتياز DBA.

وظائف مدير قاعدة المعطيات The DBA Roles

توجد مجموعة من الوظائف المتاحة لمدير قاعدة المعطيات، أهمها:

١-**DBA**: تمتلك هذه الوظيفة أغلب امتيازات ووظائف أوراكل. عند منح هذه الامتياز لمستخدم ما، فإنه يستطيع إجراء أية مهمة لذلك يجب عدم منح هذه الوظيفة إلا للمستخدمين الموثوقين.

٢-**OSOPER**: هذه الوظيفة هي إحدى وظيفتين أساسيتين تتعلقان بنظام التشغيل وتمنح لحسابات خاصة تحتاج لاستيقان نظام التشغيل OS authentication. من الضروري أن تمتلك بعض الحسابات هذا الاستيقان لأنه يمكن استخدام استيقاأوراكل

Oracle authentication فقط عندما تكون قاعدة المعطيات مفتوحة، لأن أوراكل لا يتمكن من التحقق من امتيازات المستخدمين عندما تكون قاعدة المعطيات مغلقة.

تسمح وظيفة OSOPER للمستخدم بإجراء العمليات التالية:

- *STARTUP and SHUTDOWN*
- *ALTER DATABASE MOUNT*
- *ALTER DATABASE OPEN*
- *ALTER DATABASE BACKUP*
- *ALTER DATABASE RECOVER*
- *ALTER DATABASE ARCHIVE LOG*

٣-**OSDBA**: تتضمن هذه الوظيفة الامتيازات الممنوحة للوظيفة السابقة OSPER مع بعض الامتيازات الإضافية وتتضمن الأمر CREATE DATABASE إضافةً إلى جميع امتيازات النظام مع الخيار ADMIN OPTION والذي يسمح للمستخدم بمنح هذه الامتيازات لمستخدمين آخرين أو لوظائف أخرى.

استخدام الأداة Oracle Enterprise Manager

تعتبر الأداة Oracle Enterprise Manager أحد الأدوات الجديدة التي أصدرتها شركة أوراكل للسماح لمدير قاعدة المعطيات DBA بإدارة ممثلي أوراكل باستخدام واجهات رسمية، مما يسمح بإظهار المعلومات بشكل أبسط مع دلالة أكثر.

تتكون الأداة Enterprise Manager من جزأين رئيسيين هما:

- *Graphical console*
- *Intelligent agents*

ويعتبر Enterprise Manager console الأداة الرسومية التي تسمح لك بإدارة هيئات أوراكل بشكل رسومي. يقوم الكونسول بالاتصال الأنظمة المتعددة التي يقوم بإدارتها من خلال الوكيل الذكي intelligent agents الذي يعمل على هذه الأنظمة.

يسمح الوكيل الذكي للكونسول بالاتصال مع هيئات أوراكل، وهي تستخدم بروتوكول SNMP (Simple Network Management Protocol). من أجل أخذ الطلبات من الكونسول وربط هذه الطلبات مع نظام أوراكل.

يسمح Enterprise Manager لمدير القاعدة DBA بإجراء المهام التالية:

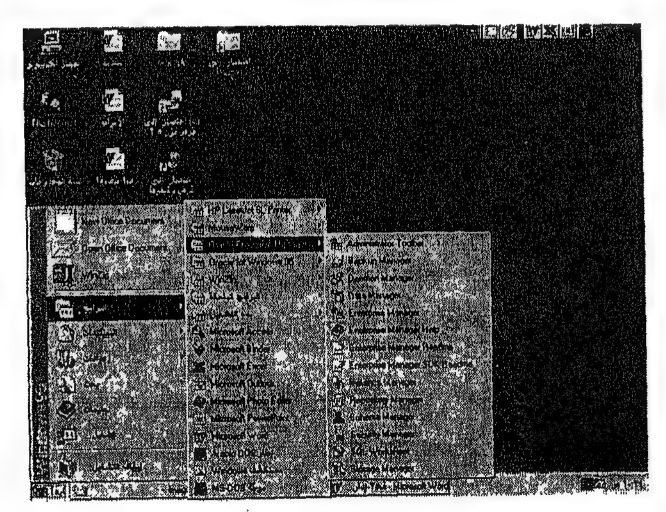
- ♦ تحسين وإدارة قواعد معطيات أوراكل.
- ♦ توزيع البرمجيات للزبائن والخدمات.
- ♦ مراقبة الأحداث الواردة من عدة هيئات.
- ♦ إجراء عمليات النسخ الاحتياطي والاسترداد من موقع وحيد.
- ♦ إجراء عمليات مدير القاعدة DBA القياسية كإدارة المستخدمين.

وتعتبر أداة Enterprise Manager أحد أقوى أدوات أوراكل وأكثرها مرونة والتي تساعدك في إجراء الكثير من مهام مدير القاعدة اليومية.



من الأسهل بالطبع إدارة نظام أوراكل من خلال الواجهة الرسومية لكن تبقى هناك العديد من الأماكن التي يفضل فيها استخدام سطر الأوامر، لذلك سنحاول في كتابنا هذا شرح الطريقتين في كل أوامر وعمليات أوراكل التي سنقوم بشرحها.

عندما تقوم بفتح مجموعة أدوات Oracle Enterprise Manager، تظهر مجموعة من الأدوات المساعدة على إدارة أوراكل (انظر الشكل ١٦-١) وهي:



الشكل ١٦-١

☆ **Storage Manager**: وهي أداة مساعدة لإدارة الفضاءات الجدولية وملفات المعطيات ومقاطع التراجع.

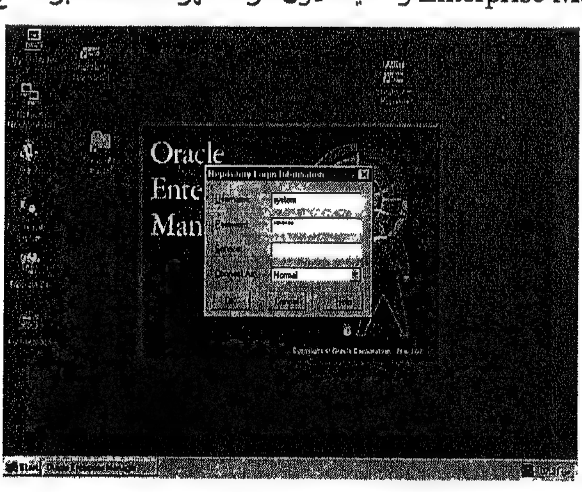
☆ **SQL Worksheet**: أداة مساعدة لتنفيذ تعليمات SQL.

- ☆ **Security Manager:** أداة مساعدة لإدارة المستخدمين والامتيازات والوظائف والتشكيلات الجانبية.
- ☆ **Schema Manager:** أداة لإدارة عناصر المخطط كالجداول والمشاهد والسلاسل والمرادفات والفهارس والتجمّعات وغيرها.
- ☆ **Replication Manager:** أداة لإدارة مخازن Enterprise Manager.
- ☆ **Instance Manager:** وهي أداة لإدارة هيئات أوراكل.
- ☆ **Data Manager:** وهي أداة لإدارة المعطيات من أجل القيام بعمليات التصدير والاستيراد وشحن المعطيات.
- ☆ **Backup Manager:** تستخدم هذه الأداة لإجراء عمليات النسخ الاحتياطي والاسترداد.

توصيف *Enterprise Manager*

يتم تركيب Enterprise Manager كجزء من إجراء تركيب Oracle Server أو كجزء من إجراء تركيب administrator client.

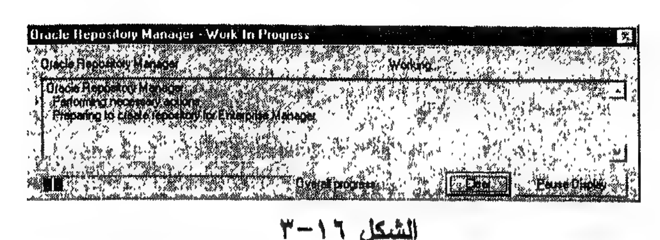
بعد أن تقوم بتركيب Enterprise Manager وتشغيله لأول مرة تظهر لك نافذة البرنامج الرئيسية والتي تطلب منك إدخال اسم الحساب وكلمة المرور والخدمة ونوع الاتصال كما في الشكل ١٦-٢(في حال كنت تقوم بتشغيل البرنامج على المخدم فلا داعي لإدخال اسم الخدمة). بعدها تظهر لك نافذة تخبرك بأن المخازن repositories الخاصة ضمن Enterprise Software Manager غير مثبتة.



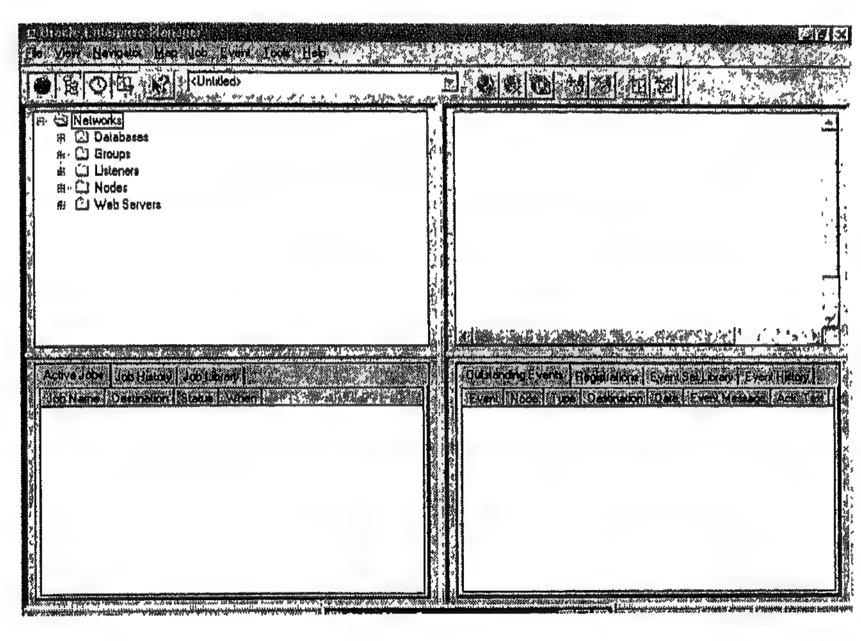
الشكل ١٦-٢

عندما تنقر على زر OK، يقوم Enterprise Manager ببناء المخازن عند أول عملية تشغيل لهذه الأداة (انظر الشكل ١٦-٣) وتأخذ هذه العملية بعض الوقت.

عند الانتهاء من بناء المخازن ، تظهر نافذة Enterprise Manager والتي تحتوي على أربع لوحات رئيسية (كما في الشكل ١٦-٤).



الشكل ١٦-٣



الشكل ١٦-٤

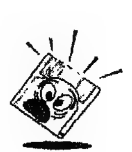
اللوحات الأربع الظاهرة في الشكل السابق هي:

١- **لوحة المستعرض Navigator Pane:** تقوم بإظهار قائمة بنمط شجرة للعناصر المختلفة المتاحة للـ Enterprise Manager. وتستطيع من خلال هذه القائمة إجراء العديد من العمليات الإدارية.

٢- لوحـــة الخـــريطة **Map Pane**: تكـــون فارغة في البداية، إلا أنها تسمح لك بإنشاء واجهـــة جغـــرافية للأنظمـــة التي ستتم إدارتها كذلك الانتقال إلى المواقع المختلفة وإدارة الأنظمة من بعد.

٣- لوحة العمل **Job Pane**: تسمح لك هذه اللوحة بمشاهدة وإدارة الأعمال المختلفة التي تعمـــل في النظام. تستطيع من خلال هذه اللوحة جدولة الأعمال المنفذة على العقد المختلفة في النظام وفي مختلف الأوقات.

٤- لوحـــة الأحداث **Event Pane**: تستخدم هذه اللوحة لرؤية أحداث النظام التي تجري في أية عقدة تقوم بإدارتها من خلال الشاشة.



عـــندما يتم تشغيل Enterprise Manager، فإنه يتم تشغيل شريط أدوات المدير Administrator toolbar، يســمح لك شريط الأدوات هذا بالوصول السريع إلى أدوات Enterprise Manager.

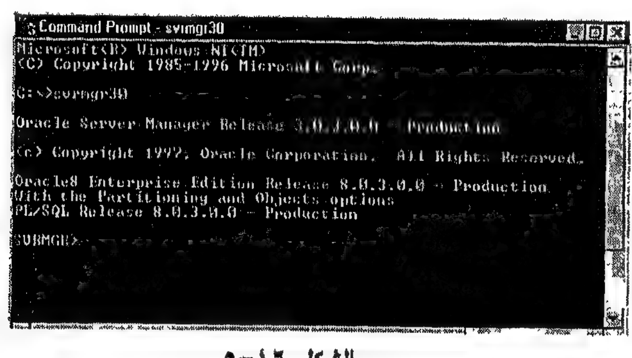
استخدام الأداة Server Manager

تمكـــنك هـــذه الأداة مـــن الـــتعامل مـــع هيـــئة أوراكـــل اعـــتماداً على واجهة محارف character_based interface والكتابة من خلال أسطر الأوامر Command lines.

وتســـتطيع تشـــغيل Server Manager مباشرة من خلال سطر أوامر Windows NT بكتابة اسم البرنامج على الشكل:

C: > Svrmgr30

تظهـــر لـــك نافذة Server Manager كمـــا في الشكل١٦-٥: تستطيع من خلال هـــذه النافذة كتابة



الشكل ١٦-٥

تعليمات SQL مباشرة.

فمــثلاً تستطيع الاتصال مع حساب system/manager ومن ثم إظهار جميع قيم الوسطاء كما في الشكل ١٦-٦:

```
Command Prompt - svrmgr30
db_block_lru_statistics              boolean FALSE
db_block_max_dirty_target            integer 4294967294
db_block_size                        integer 2048
db_file_multiblock_read_count        integer 16
delayed_logging_block_cleanouts      boolean TRUE
hash_multiblock_io_count             integer 8
log_block_checksum                   boolean FALSE
SVRMGR> connect system/manager
Connected.
SVRMGR> show parameters block;
NAME                                 TYPE    VALUE
------------------------------------ ------- ------------------------------
db_block_buffers                     integer 550
db_block_checkpoint_batch            integer 8
db_block_checksum                    boolean FALSE
db_block_lru_extended_statistics     integer 0
db_block_lru_latches                 integer 1
db_block_lru_statistics              boolean FALSE
db_block_max_dirty_target            integer 4294967294
db_block_size                        integer 2048
db_file_multiblock_read_count        integer 16
delayed_logging_block_cleanouts      boolean TRUE
hash_multiblock_io_count             integer 8
log_block_checksum                   boolean FALSE
SVRMGR>
```

الشكل ١٦-٦




الباب الخامس



١٧ . إدارة هيئة أوراكل.

١٨ . توصيف الشبكة.

١٩ . إقلاع وإطفاء قاعدة معطيات أوراكل.

١٦

إدارة هيئة أوراكل

Administrating Oracle Instance

ضمن نظام Windows NT، يجب إنشاء هيئة إقلاع أوراكل قبل أن يتم إنشاء قاعدة المعطيات. وبما أنها عبارة عن خدمة Service فإن هيئة أوراكل ضمن نظام Windows NT تختلف قليلاً عن نظرائها في الأنظمة الأخرى.

ويتم إنشاء خدمة بالاسم Oracle Service SID (حيث SID هو محدد النظام) عند إنشاء هيئة أوراكل، وتقوم هذه الخدمة بتشغيل هيئة أوراكل الافتراضي. ولا يمكنك هنا الاتصال بأوراكل أو إنشاء قاعدة معطيات دون إقلاع هذه الخدمة.

ستقوم في هذا الفصل بشرح مفصل عن العمليات الأساسية المتعلقة بهيئات أوراكل وذلك باستخدام طرق مختلفة.

إنشاء هيئة جديدة Creating New Instance

تذكر في الفصل السادس، عندما قمت بتركيب نسخة أوراكل على المخدّم فلقد طلب منك النظام تحديد كيفية إنشاء هيئة أوراكل جديد.

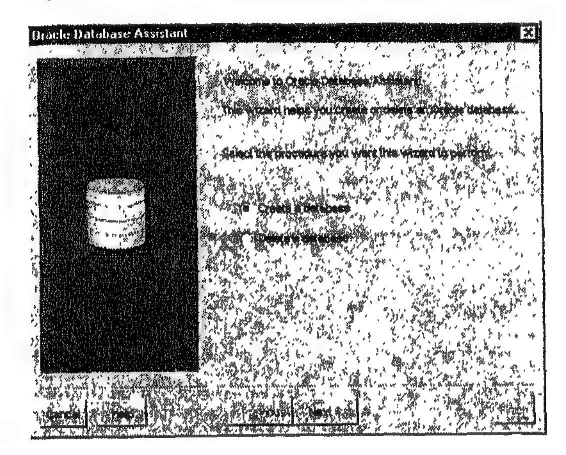
تستطيع إجراء هذه العملية باستخدام عدة طرق وهي:

إنشاء هيئة أوراكل باستخدام Oracle Database Assistant

عندما تقوم بتشغيل برنامج Oracle Database Assistant، ستظهر لك نافذة مشابهة للشكل ١٧-١ تطلب منك اختيار العمل الذي ترغب بإنجازه وهو إما إنشاء قاعدة معطيات أو حذف قاعدة.

سنقوم هنا باختيار الأمر الأول: Create a database.

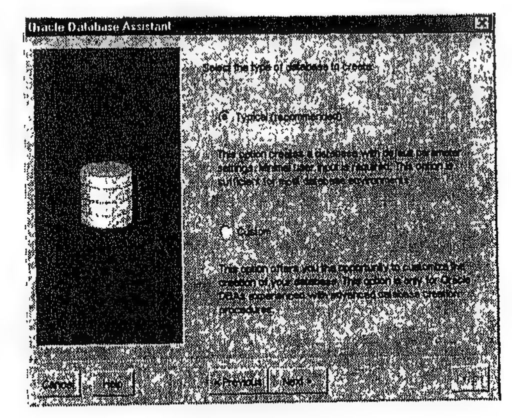
انقر على زر Next.



الشكل ١٧-١

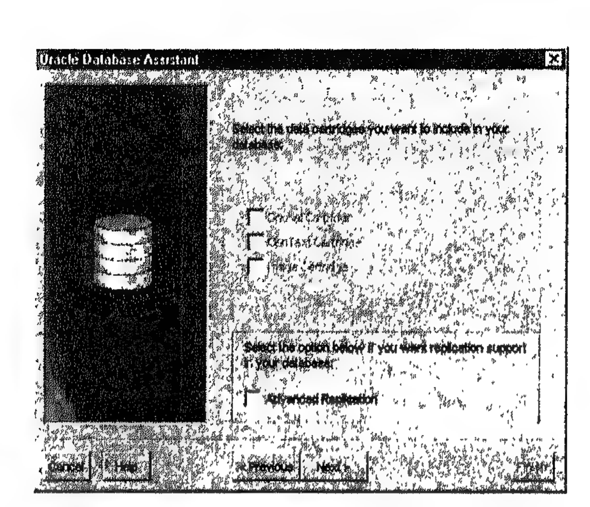
تظهر النافذة الثانية كما في الشكل ١٧-٢، تطلب منك هذه النافذة اختيار طريقة إنشاء قاعدة المعطيات إما بشكل Typical أو Custom.

اختر هنا الطريقة الأولى Typical (أما الطريقة الثانية فسيتم شرحها بالتفصيل في الفصل الثامن عشر) ثم انقر زر Next.



الشكل ١٧-٢

ستظهر لك نافذة جديدة كما في الشكل١٧-٣ تطلب منك ان كنت ترغب بإنشاء نسخة مماثلة لهذه القاعدة replication (سيتم شرح هذه العملية بالتفصيل في الفصل ٤٦) انقر زر Next دون تحديد أي من صناديق التحقق.

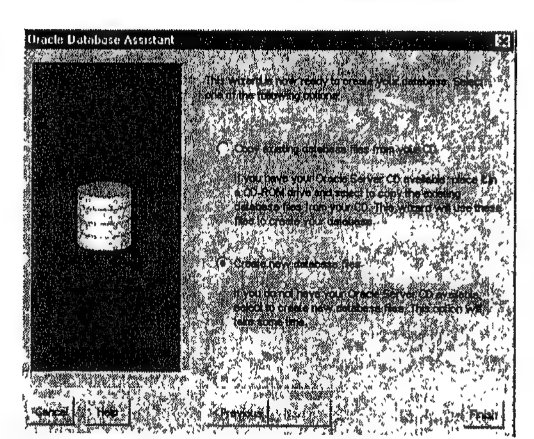


الشكل ١٧-٣

ستظهر نافذة جديدة كما في الشكل١٧-٤:

ستسألك هذه النافذة إن كنت ترغب بنسخ ملفات قاعدة المعطيات من القرص المدمج CD- ROM أو بإعادة إنشائها.

سنطلب الخيار الثاني: Create new Database files



الشكل ١٧-٤

انقر زر Finish لبدء عملية إنشاء قاعدة المعطيات، تظهر لك نافذة حوار تطلب منك التأكد من رغبتك فعلاً ببدء إنشاء القاعدة. انقر زر Yes، تظهر لك النافذة الأخيرة التي توضح لك اسم قاعدة المعطيات التي سيتم إنشاؤها ومحدد النظام SID.

انقر زر OK، لبدء عملية إنشاء القاعدة والتي تأخذ بعض الوقت.

إنشاء هيئة أوراكل من خلال سطر أوامر NT

يمكنك طلب برنامج إدارة هيئات أوراكل ORADIM80.EXE مباشرةً من خلال سطر أوامر NT والتي تأخذ الشكل:

ORADIM80 – NEW- SID sid – INTPWD password
[-MAXUSERS number]
[-STARTMOD AUTO or MANUAL]
[-PFILE pfile_name]

ORADIM 80 –NEW – SID Moh – INTPWD oracle

التعليمة السابقة تقوم بإنشاء هيئة جديدة بالاسم Moh وكلمة المرور oracle وهنا سيتم إنشاء ملف وسطاء افتراضي بالاسم initMoh.Ora، أما إقلاع الهيئة فيتم بشكل يدوي.

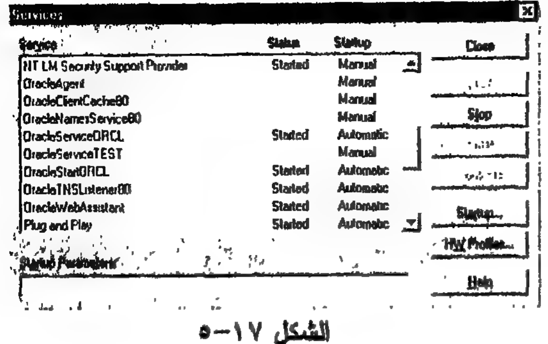
إقلاع هيئة أوراكل Starting Up An Instance

يمكن إقلاع هيئة أوراكل إما بشكل يدوي manually أو بشكل تلقائي Automatically أو من خلال الأداة الرسومية أو من خلال سطر الأوامر، سنقوم بشرح هذه الطرق المختلفة لإقلاع هيئة أوراكل:

إقلاع هيئة أوراكل بشكل تلقائي

يمكن تحديد إقلاع هيئة أوراكل بشكل تلقائي عند إقلاع نظام Windows NT، من أجل ذلك أدخل إلى لوحة التحكم Control Panel في المخدّم وافتح نافذة Service.

تظـــهر قائمـــة بالخدمـــــات Services الفعالة في النظـــام، انقر نقراً مزدوجاً على الخدمة Oracle Service SID (حيث SID محـــدد النظـــام، يمكن أن يكون مثـــلاً موافـــق لهيئة أوراكل التي تمّ إنشـــاؤها في المثال السابق على الشكل OracleServiceTEST).

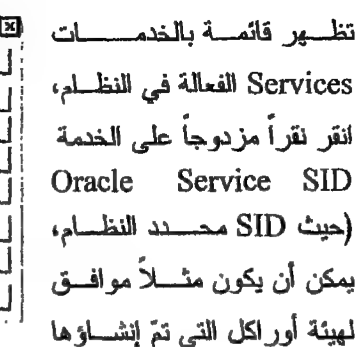


الشكل ١٧–٥

تظهر نافذة تشبه الشكل١٧–٥.

تمكنك هذه النافذة من تغيير طريقـــة إقلاع الهيئة. انقر نقراً مزدوجاً على الخدمة التي ترغب بتغيير طريقـــة إقلاعها، تظهر نافذة تشـــبه الشـــكل ١٧–٦.

تستطيع من خلال صندوق الحـــوار هذا تحديد طريقة الإقلاع إما تلقائيـــة Automatic أو يدوية Manual أو إلغاء تأهيل الخدمة Disable.



الشكل ١٧–٦

إقلاع هيئة أوراكل من خلال سطر أوامر NT

يمكن إقلاع هيئة أوراكل من خلال سطر الأوامر باستخدام البرنامجORADIM80.EXE على الشكل التالي:

ORADIM80 – STARTUP – SID sid –PFILE pfile_ name
[-USRPWD password] – STARTTYPE SRVC or inst

حيث:

- ❖ **SID**: محدد النظام.
- ❖ **Pfile-name**: اسم ملف الوسطاء.

إطفاء هيئـــة أوراكـــل SHUTTING DOWN An INSTANCE

لأن كثيراً من المعطيات التي تم إجراء تغييرات عليها في قاعدة معطيات أوراكل تســتقر في الذواكر المؤقتة لقاعدة المعطيات، كما أن الإطفاء غير المنظم لقاعدة المعطيــات قـد يحتاج إلى وقت استرداد طويل عند إعادة الإقلاع مرة ثانية.

لذلك تستطيع تجنب ذلك بإجراء إطفاء منظم لقاعدة المعطيات. وتوجد أربعة أنماط لعمليــة إطفاء القاعدة هي:

- **إطفاء نظامي Normal**: وهو أكثر عمليات الإطفاء التي ينصح بــها.ولا يمكــن لأي مستخدم جديد الدخول إلى قاعدة المعطيات، وتبقى هيئة أوراكل في حالة انتظــار حتى يقوم جميع المستخدمين بقطع الاتصال مع القاعدة قبل أن تتم عملية الإطفاء.
- **إطفاء فوري Immediate**: في هذا النوع، يتم إنهاء جميع التحويــلات الفعالــة حالياً وأي تحويل غير مثبت uncommited transaction يتم اســترجاعه rolled back.

كما يتم قطع الاتصال بين جميع مستخدمي القاعدة وبين هيئة أوراكل فوريــاً ثــم تتــم عملية الإطفاء بعد ذلك.

- **إطفاء التحويلات Transactional**: يسمح لك هذا الخيار بإنهاء جميع التحويلات الفعالة، يتم بعدها قطع الاتصال وإطفاء الهيئة.
- **الإحباط Abort**: يتم في هذا النوع من الإطفــاء، قطــع الاتصــال مــع جميــع المستخدمين وإطفاء هيئة أوراكل بشكل فوري حتى لو كانت هناك بعض المشاكل. هذا النوع من الإطفاء يؤدي إلى طلب استرداد قاعدة المعطيات عند أول إقلاع للهيئة بعــد ذلك.

إطفاء هيئة أوراكل تلقائياً Shutting Down The instance Automatically

كما رأينا سابقاً فإنك تستطيع اختيار الإقلاع التلقائي لقاعدة المعطيات ضمن مخدّم NT. في هذه الحالة يقوم المخدّم بالإطفاء التلقائي للقاعدة عند إطفاء المخدّم.

إطفاء هيئة أوراكل من خلال سطر أوامر NT

يمكن إطفاء هيئة أوراكل من خلال سطر الأوامر باستخدام برنامج ORADIM80.EXE على الشكل التالي:

ORADIM80 – SHUTDOWN – SID sid –PFILE pfile –name
[-USRPWD password] – SHUTTYPE SRVE Or inst
SHUTMODE a, I, n

حيث:

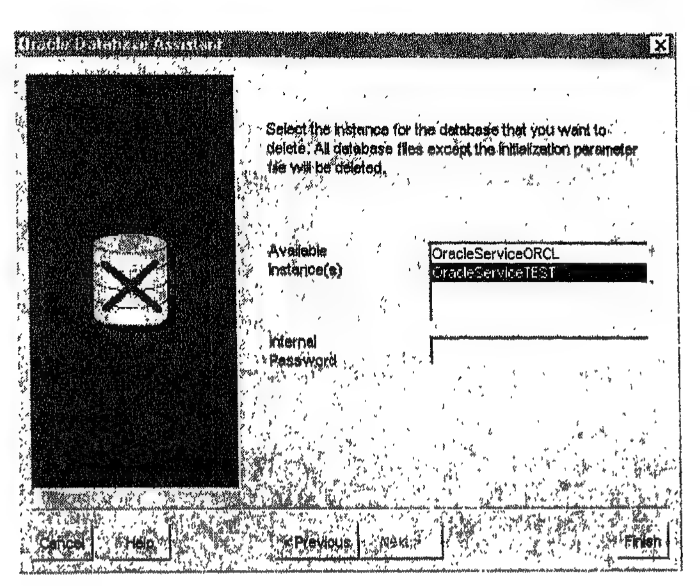
a = abort
I =immediate
N = normal

حذف هيئة أوراكل Deleting an Instance

يمكن حذف هيئة أوراكل باستخدام إحدى الطريقتين التاليتين:

حذف هيئة أوراكل باستخدام Oracle Database Assistant

عند تشغيل برنامج Oracle Database Assistant تظهر نافذة مشابهة للشكل١٧-١.
اطلب الخيار Delete a database ثم انقر زر Next، تظهر نافذة مشابهة للشكل ١٧-٧.
حدّد في هذه النافذة الهيئة التي ترغب بحذفها ثم أدخل كلمة مرور حساب Internal، أخيراً انقر زر Finish.

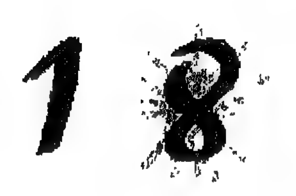


الشكل ١٧-٧

حذف هيئة أوراكل من خلال سطر أوامر NT

يمكنك حذف هيئة أوراكل من خلال سطر الأوامر باستخدام برنامج ORADIM80.EXE على الشكل:

ORADIM80 –DELETE – SID sid
Or
ORADIM80 –DELETE – SRVC service



توصيف الشبكة
Configuring Network

بعد إنشاء هيئة أوراكل يجب توصيف الشبكة حتى تستطيع الاتصال مع قـــاعدة المعطيات الموجودة على المخدّم. وهذا يفيد من ناحيتين:

* للسماح لإجراء بعيد remote process بالاتصال مع قـــاعدة المعطيـــات ويتـــم التعديل على الملف LISTNENER.ORA.
* للسماح بالاتصال مع الهيئة من خلال رديف، حيث يمكنك بهذه الطريقة الاتصـــال مع عدة قواعد معطيات بمحددات نظام SID مختلفة وذلك بالاتصال مع الخدمة التـــي تدل على SID.

توجد طريقتان لتوصيف خدمة الخدمة:

١-من خلال مولد طبولوجية الشبكة Network Topology Generator الموجـــود في Enterprise Manager.

٢-من خلال معالج توصيف شبكة أوراكـل Oracle Network Configuration Wizard الموجود في ORACLE NT.

توصيف ملف LISTENER.ORA

يتم إنشاء الملف LISTENER.ORA لكل بروتوكول شبكة أثناء وقت التثبيت. يقوم هـــذا الملف بتعريف بروتوكولات الشبكة كذلك محددات أوراكل SIDs التي يمكن الوصول إليها من خلال المستمع، افتراضياً فإن SID المعرف في هذا الملف هو ORCL.

الشكل الابتدائي لملف LISTENER.ORA يشبه:

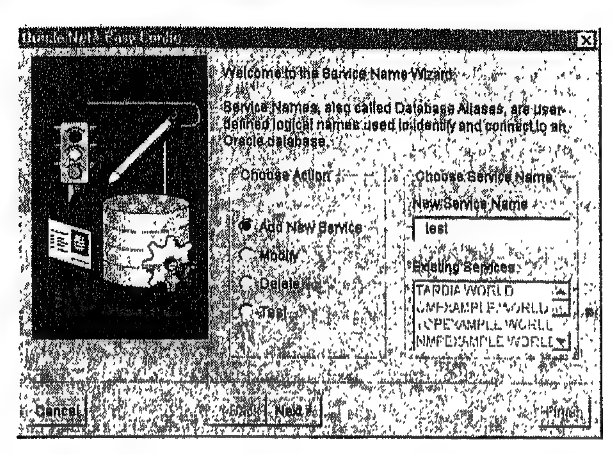
```
LISTENER =
 (ADDRESS_LIST =
     (ADDRESS=
       (PROTOCOL= IPC)
       (KEY= oracle.world)
     )
     (ADDRESS=
       (PROTOCOL= IPC)
       (KEY= ORCL)
     )
     (ADDRESS=
       (PROTOCOL= IPC)
       (KEY= EXTPROC0)
     )
     (ADDRESS=
       (COMMUNITY= NMP.world)
       (PROTOCOL= NMP)
       (SERVER= TARP_SERVER)
       (PIPE= ORAPIPE)
     )
     (ADDRESS=
       (PROTOCOL= TCP)
       (Host= tarp_server)
       (Port= 1521)
     )
```

```
    (ADDRESS=
     (PROTOCOL= TCP)
     (Host= tarp_server)
     (Port= 1526)
    )
    (ADDRESS=
     (PROTOCOL= TCP)
     (Host= 127.0.0.1)
     (Port= 1521)
    )
    (ADDRESS=
     (PROTOCOL= SPX)
     (Service= tarp_server_lsnr)
    )
 )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = 0
SID_LIST_LISTENER =
 (SID_LIST =
  (SID_DESC =
   (GLOBAL_DBNAME = tarp_server)
   (SID_NAME = ORCL)
  )
  (SID_DESC =
   (SID_NAME = extproc)
   (PROGRAM=extproc)
  )
 )
PASSWORDS_LISTENER = (oracle)
```

يمكنك إضافة محدد نظام SID جديد إلى قائمة المحددات، انسخ فقط الفقرة من LISTENER.ORA التي تقوم بتوصيف المحدد SID والصقها في هذا الملف بعد تعريف محدد النظام.

توصيف الشبكة باستخدام الأداة *ORACLE Net8 Easy Config*

من أجل إدارة الملف TNSNAMES.OR، يمكنك استخدام الأداة Oracle Net 8 Easy Config على الشكل التالي:

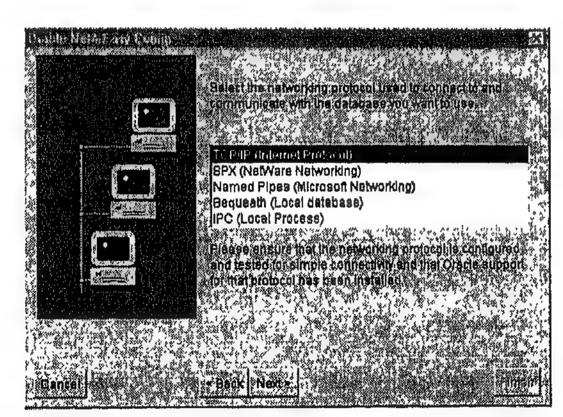


الشكل ١٨-١

١- عندما يتم تشغيل هذه الأداة تظهر نافذة حوار تمكنك من إنشاء خدمة جديدة أو تعديلها أو حذفها أو اختبارها، انظر الشكل ١٨-١.

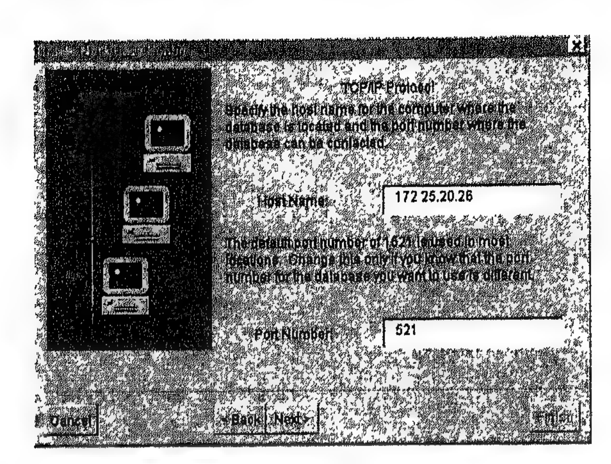
لإضافة خدمة جديدة اختر Add New Service واكتب اسم هذه الخدمة في الحقل New Service Name، ثم انقر زر Next.

٢-تظهر نافذة جديدة تطلب تحديد اسم بروتوكول الشبكة الخاص بها. اختر اسم البروتوكول المطلوب ثم انقر زر Next. انظر الشكل ١٨-٢.



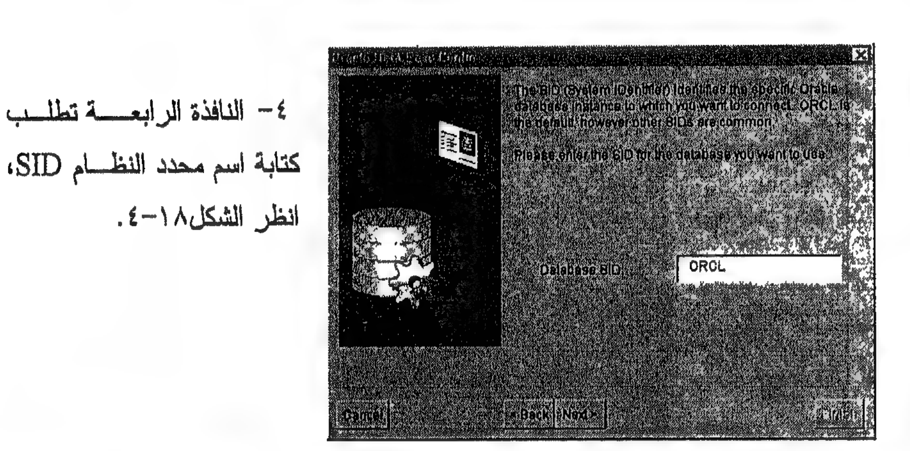
الشكل ١٨-٢.

٣-تطلب النـــافذة الثالثة اسم الحاســب المضيــف HOST NAME، اكتب هنا اسم الحاسب المخدّم أو IP الخاص بـــه، انظر الشـــكل ١٨-٣.



الشكل ١٨-٣

٤- النافذة الرابعـــة تطلـــب كتابة اسم محدد النظـــام SID، انظر الشكل١٨-٤.



الشكل ١٨-٤

الشكل ١٨-٥

٥- تتيح لك النافذة الخامســـة إمكانية اختبار هذه الخدمـــة، انظر الشكل١٨-٥.

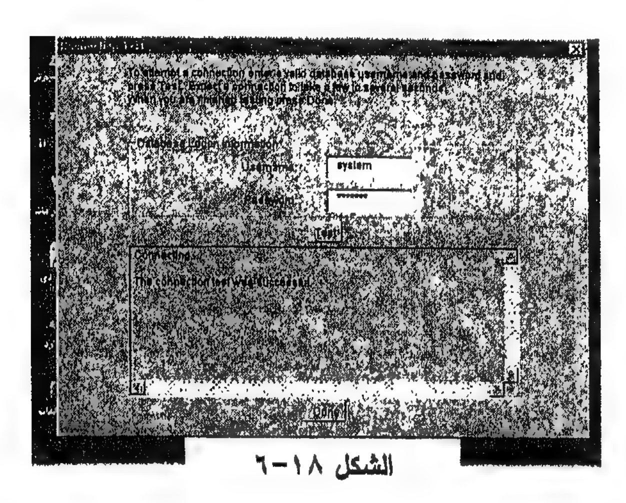
انقر زر Test Service، تظـــهر نافذة تشبه الشكل١٨-٦.

حدد اسم المستخدم وكلمة السر ثم انقر زر Test.

فإذا ظهرت الرسالة:

The Connection test was successful

فهذا يعني أنك قد نجحت في إنشاء الخدمة.



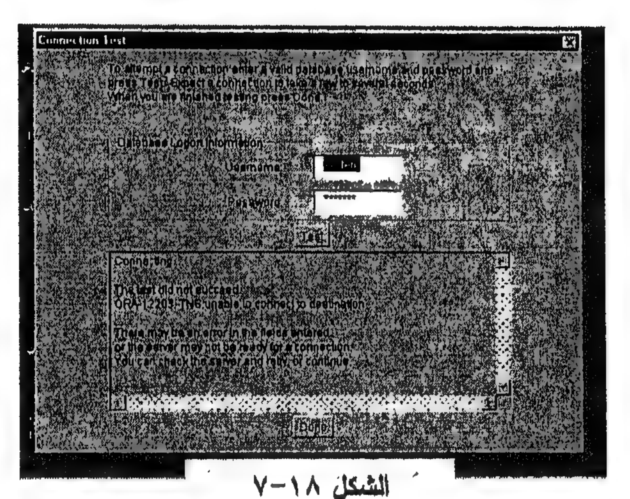
الشكل ١٨-٦

أما إذا ظهرت الرسالة:

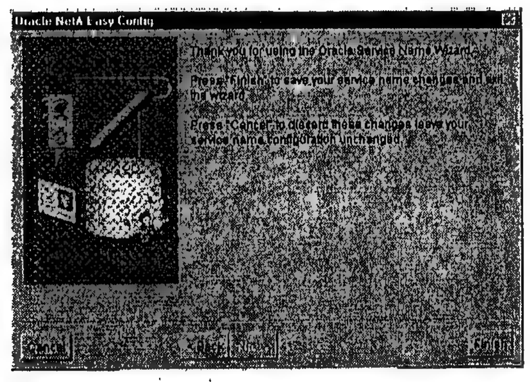
The test did not succeed

فهذا يعني أنك فشلت في إنشاء الخدمة، قم بالعودة إلى النوافذ السابقة وتصحيح الخطأ الموجود.

٦-انقر زر Done في حال نجاح إنشاء الخدمة ثم انقر زر Finish الموجود في الشكل وذلك لإنهاء إنشاء الخدمة (انظر الشكل ١٨-٨).



الشكل ١٨-٧



الشكل ١٨-٨





إقلاع وإطفاء قاعدة معطيات أوراكل

Startup and Shutdown a Database

سنتعلم فــي هذا الفصل الإجراءات الأساسية المستخدمة عند إقلاع قاعدة معطيات أوراكل وعند إطفائها، والمراحل التي تمرّان بها.

إقلاع قاعدة معطيات أوراكل Startup a Database

عملية إقلاع نظام أوراكل تتضمن العديد من الإجراءات هي :

* إقلاع هيئة **Starting an Instance**: وهي عملية حجز منطقة النظام العامة SGA وإقلاع إجراءات الخلفية Background Processes. هذه العملية مرتبطة بعملية تركيب Mount قاعدة المعطيات، فإذا تمّ إقلاع الهيئة فإنه لايتم إجراء أي ارتباط بين قاعدة المعطيات وبين SGA وإجراءات الخلفية.

قبل إنشاء هيئة، يقوم أوراكل بقراءة ملف الوسطاء Parameter File الذي يحدد كيفية إقلاع الهيئة، وهو يحتوي على وسطاء تحدد حجم ذاكرة SGA، واسم القاعدة التي ستتصل الهيئة بها.

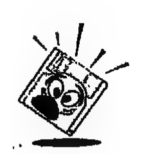
يمكن إقلاع هيئة بالنمط المقيد restricted mode، حيث يتم تحديد الاتصالات عند فتح القاعدة مع حسابات المستخدمين الذي يمتلكون الامتياز RESTRICTED SESSION.



في بعض الظروف الاستثنائية، وفي حال عدم إطفاء هيئة قاعدة معطيات بشكل سليم تماماً (كأن تبقى إحدى إجراءات الهيئة فعالة not killed) فإنه يمكن حلّ هذه المشكلة بقتل kill جميع إجراءات أوراكل المتبقية من الهيئة السابق وإعادة تشغيل الهيئة الجديدة.

* تركيب قاعدة معطيات **Mounting a Database**: وهي عملية ربط قاعدة معطيات مع الهيئة التي تمّ إقلاعها مسبقاً، وتبقى قاعدة المعطيات مغلقة بعد تركيبها حيث يمكن لمدير القاعدة الدخول إليها لإنجاز بعض عمليات الصيانة المحددة فقط.

عند تركيب القاعدة، تقوم الهيئة بالبحث عن ملفات التحكم (المحددة بالوسيط CONTROL_FILES في ملف الوسطاء) وفتحها من أجل الحصول على أسماء ملفات المعطيات Data Files وملفات الإرجاع Redo Log Files.



يمكن تشغيل أوراكل بنمط Exclusive أو Parallel، حيث يتم إقلاع عدة هيئات بشكل متزامن لتركيب نفس القاعدة.

* **فتح قاعدة معطيات Open a Database:** وهي عملية جعل قاعدة المعطيات متاحة لإجراء العمليات الاعتيادية عليها، حيث يمكن لأي مستخدم مرتبط الاتصال بالقاعدة والوصول إلى معلومات هذه القاعدة.

بعد قيام الهيئة بفتح القاعدة ، تحاول جلب مقطع أو أكثر من مقاطع التراجع Rollback segments.

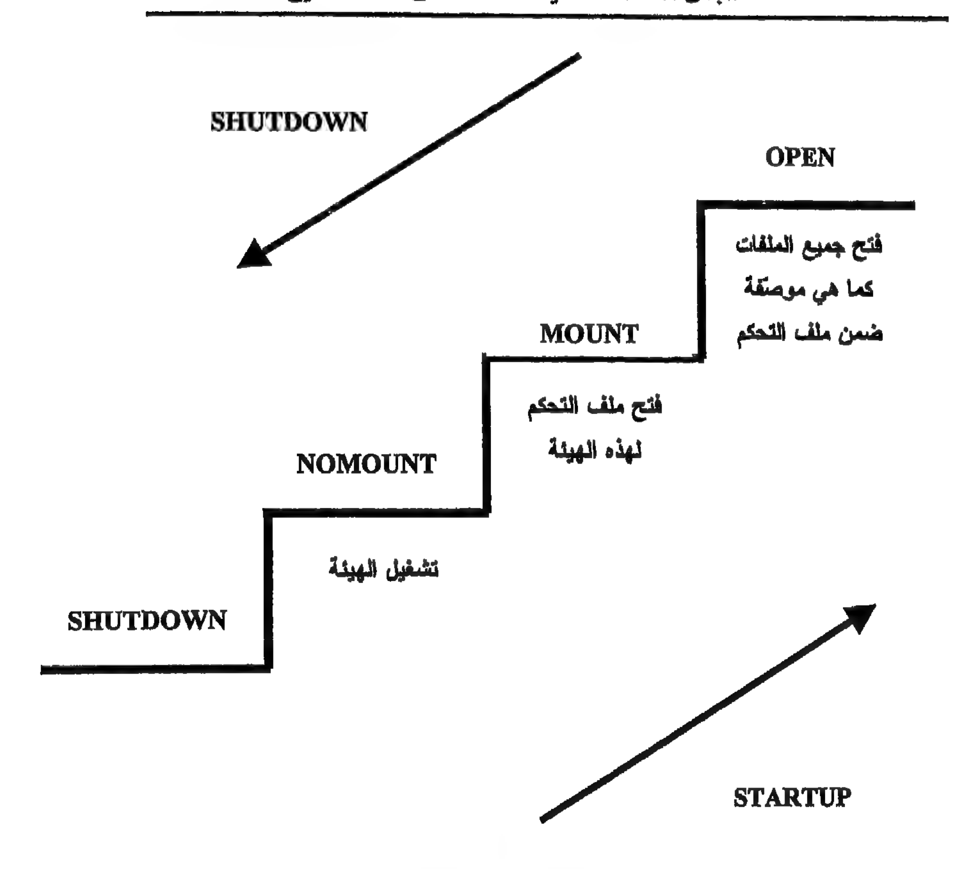


في حال إطفاء قاعدة المعطيات بشكل غير نظامي، تتم عملية إجراء استرداد recovery تلقائية لهذه القاعدة عند فتحها.



يبين الجدول التالي وصف كل حالة من حالات الإقلاع :

حالة الإقلاع	وصف الحالة
NOMOUNT	تستخدم لإنشاء قاعدة معطيات.
MOUNT	تستخدم لتعديل بنى الملفات وتعديل محتوى ملف التحكم.
OPEN	تجعل قاعدة المعطيات متاحة لكل المستخدمين.



من أجل إقلاع قاعدة معطيات Startup a database أو إقلاع هيئة Startup an Instance تستطيع طلب تعليمة الإقلاع Startup باستخدام أحدى الطريقتين التاليتين:

إقلاع قاعدة معطيات أوراكل من خلال Server Manager

تستطيع من خلال الأداة Server Manager استخدام تعليمة الإقلاع على الشكل التالي:

STARTUP
[PFILE = Parameter_file]
[MOUNT or NO MOUNT]
[OPEN]
[EXCLUSIVE]
[RESTRICT]
[FORCE]
[PARALLEL / SHARED]

حيث:

- **PFILE =Parameter_ file**: لتحديد اسم ملف الوسطاء الذي سيتم استخدامه ويدلّ بشكل افتراضي على الملف initORCL.Ora.
- **NOMOUNT**: لإقلاع الهيئة دون تركيب القاعدة.
- **MOUNT [database]**: لإقلاع الهيئة وتركيب قاعدة معطيات بالاسم database، لكن دون فتحها.
- **OPEN [database]**: لإقلاع وتركيب وفتح قاعدة معطيات بالاسم database، وهذا الوسيط هو وسيط افتراضي.
- **FORCE**: لإقلاع الهيئة حتى لو كانت هناك بعض المشاكل.
- **RESTRICT**: لإقلاع الهيئة بالنمط المقيد، حيث يمكن فقط للمستخدمين الذين يمتلكون الامتياز RESTRICT SESSION الاتصال بقاعدة المعطيات.
- **EXCLUSIVE**: للسماح فقط لهيئة وحيدة بتركيب قاعدة المعطيات في بيئة مخدّم متوازي Parallel-Server Environment.

توجد إذاً عدة طرق للإقلاع:

١- إقلاع هيئة بدون تركيب قاعدة المعطيات Starting an Instance Without Mounting a Database:

تستخدم هذه الطريقة فقط عند بناء قاعدة المعطيات ويمكن من خلال بطلب تعليمة Startup مع الخيار Nomount.



STARTUP NOMOUNT sales PFILE=INITSALE. ORA

٢-إقلاع هيئة وتركيب قاعدة المعطيات Starting an Instance and Mounting a Database:

يمكنك إقلاع هيئة وتركيب قاعدة معطيات بدون فتح هذه القاعدة وذلك لإجراء بعض عمليات الصيانة الخاصة مثلاً:

- تغيير أسماء ملفات المعطيات.
- إضافة، حذف وتغيير أسماء ملفات الإرجاع redo log files.
- تأهيل Enabling وإلغاء تأهيل Disabling خيار أرشفة ملفات الإرجاع Redo log.
- إجراء عمليات استرجاع لكامل قاعدة المعطيات.

يمكن القيام من خلال ذلك بطلب تعليمة Startup مع الخيار MOUNT.



STARTUP MOUNT sales PFILE=INITSALE. ORA

٣-إقلاع هيئة مع تركيب وفتح قاعدة معطيات Starting an Instance and Mounting and Opening a Database:

وهي عملية الإقلاع الاعتيادية، وهي تسمح لأي مستخدم للقاعدة بالاتصال معها وإجراء عمليات الوصول إلى معطيات هذه القاعدة.

يمكن القيام عن طريق تعليمة Startup مع الخيار OPEN.

٤- تقييد الوصول إلى قاعدة المعطيات عند الإقلاع Restricting Access To a Database at Start:

ربما ترغب بإقلاع هيئة ومن ثم تركيب وفتح قاعدة معطيات وفق نمط مقيد restricted mode وذلك لجعل القاعدة متاحة فقط لمدير القاعدة كي يقوم بإحدى المهام التالية:

- إجراء إصلاحات على البنية كإرجاع بناء الفهارس.
- إجراء عمليات التصدير Export والاستيراد Import للمعطيات.

- إجراء عمليات شحن للمعطيات باستخدام SQL*Loader.
- منع المستخدمين العاديين من الوصول إلى المعطيات بشكل مؤقت.

يمكن إجراء هذه العملية بطلب تعليمة STARTUP مع الخيار RESTRICT.



STARTUP OPEN sales PFILE=INITSALE. ORA EXCLUSIVE RESTRICT

٥- إجبار هيئة على الإقلاع Forcing an Instance to Start:

في بعض الظروف الاستثنائية، قد تواجه مشاكل عديدة عند محاولة إقلاع هيئة قاعدة معطيات.

لذلك يجب عدم إجبار الهيئة على الإقلاع إلا في أحدى الحالات التالية:

- لا يمكن إطفاء الهيئة الحالي بنجاح من خلال استخدام خيارات الإطفاء Normal أو Immediate.
- حدوث مشاكل عديدة عند إقلاع الهيئة.

ويمكن حل هذه المشاكل بإقلاع هيئة جديد عن طريق استخدام تعليمة STARTUP مع الخيار FORCE.

٦- إقلاع هيئة وتركيب قاعدة معطيات وبدء عملية استرجاع كاملة Starting an Instance, Mounting a Database and Starting Complete Media Recovery:

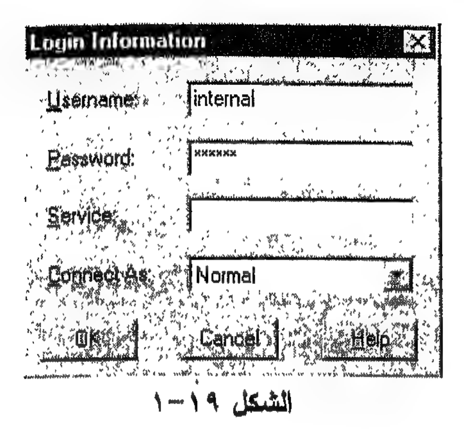
إذا عرفت بأن عملية استرداد الوسائط media recovery مطلوبة وتستطيع إقلاع هيئة وتركيب قاعدة المعطيات على الهيئة والحصول بشكل تلقائي على إجراء الاسترداد باستخدام تعليمة STARTUP مع الخيار RECOVER.

إقلاع قاعدة معطيات أوراكل من خلال Enterprise Manager

بعد إنشاء هيئة الإقلاع bootstrap instance ، يمكنك استخدام الأداة Instance Manager الموجودة في Enterprise Manager لإقلاع وإطفاء قاعدة معطيات أوراكل.

انقر على زر Instance Manager، سيظهر لك صندوق حوار كما في الشكل١٩-١.

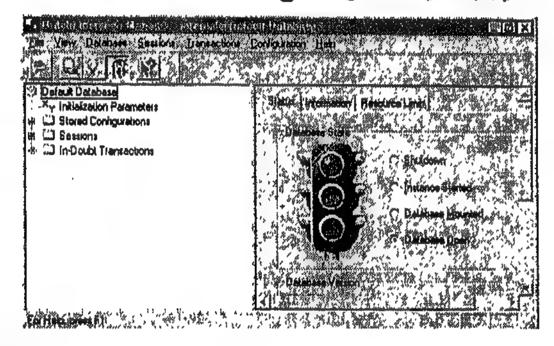
تطلب منك هذه النافذة إدخال اسم المستخدم و كلمة المرور واسم الخدمة وطريقة الاتصال، هنا يجب عليك الدخول على حساب internal



الشكل ١٩-١

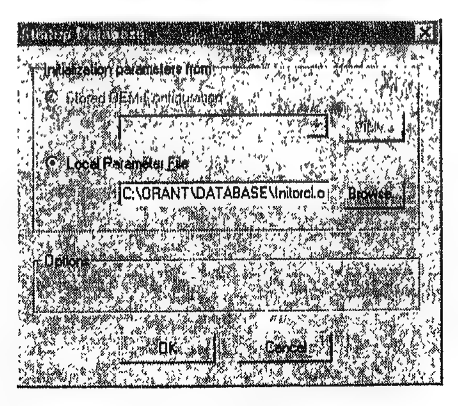
بعد إجراء عملية الاتصال، ستظهر نافذة إقلاع الهيئة في الطرف اليميني من النافذة، كما في الشكل١٩-٢.

يمكنك هنا تحديد خيارات الإقلاع وهي:



الشكل ١٩-٢

- Instance Started: لإقلاع الهيئة دون تركيب القاعدة.
- Database Mounted: لإقلاع الهيئة مع تركيب القاعدة لكن دون فتحها.
- Database Open: حيث يتم هنا إقلاع وتركيب وفتح قاعدة المعطيات.



الشكل ١٩-٣

انقر زر Apply، تظهر نافذة جديدة تشبه الشكل ١٩-٣، تطلب منك تحديد ملف وسطاء إقلاع الهيئة.

تعديل إمكانيات الوصول إلى قاعدة معطيات Altering Database Availability

يمكن جعل قاعدة معطيات متاحة جزئياً للمستخدمين، فمثلاً يمكن فتح قاعدة معطيات مغلقة ثم تركيبها بحيث يمكن للمستخدمين الاتصال بهذه القاعدة والتعامل معها.

نستطيع كتابة الشكل العام لتعليمة تعديل امكانيات الوصول إلى قاعدة معطيات:

ALTER DATABASE [database]
MOUNT [EXCLUSIVE | PARALLEL] |
OPEN [RESETLOGS | NORESETLOGS]

حيث:

- **database**: اسم قاعدة المعطيات التي سيتم تعديلها.
- **MOUNT**: لتركيب القاعدة دون فتحها للمستخدمين العاديين، حيث يقوم مدير القاعدة في هذه الحالة بالقيام ببعض المهام.
- **EXCLUSIVE**: وهو الخيار الافتراضي، ويسمح للهيئة الحالية فقط بالاتصال مع قاعدة المعطيات.
- **PARALLEL**: للسماح لعدة هيئات بالاتصال مع قاعدة المعطيات.
- **OPEN**: للسماح للمستخدمين العاديين بالاتصال مع قاعدة المعطيات.

أما الخياران RESERTLOGS و NORESETLOGS فسيتم شرحهما في الفصل الخاص بالنسخ الاحتياطي والاسترداد Oracle Backup and Recovery.

١. تركيب قاعدة معطيات على هيئة Mounting a Database To an Instance:

عندما تحتاج لإجراء بعض عمليات الإدارة الخاصة على قاعدة معطيات، يجب أن يكون قد تم إقلاع وتركيب هذه القاعدة على هيئة مع الإبقاء عليها مغلقة. وعند القيام بتركيب القاعدة، يمكن تحديد طريقة التركيب هذه بشكل استثنائي Exclusively على هذا الهيئة فقط، أو تركيب القاعدة بحيث يمكن إجراء عملية التركيب بشكل متزامن Concurrently

على الهيئات الأخرى. من أجل تركيب قاعدة معطيات الهيئة ثم إقلاعه بشكل مسبق استخدم تعليمة ALTER DATABASE مع الخيار MOUNT.



ALTER DATABASE MOUNT ;

٢. فتح قاعدة معطيات مغلقة **Opening a Closed Database**: يمكنك جعل قاعدة معطيات مركبة لكن مغلقة متاحة للمستخدمين العاديين بفتح هذه القاعدة.
للقيام بذلك استخدم تعليمة ALTER DATABASE مع الخيار OPEN.



ALTER DATABASE OPEN;

إطفاء قاعدة معطيات Shutting Down a Database

من أجل إطفاء قاعدة معطيات وهيئة قاعدة، يجب عليك أولاً الاتصال بالحساب INTERNAL ومن ثم إجراء إحدى طرق الإطفاء التالية:

إطفاء قاعدة معطيات أوراكل باستخدام Server Manager

يمكن استخدام الأداة *Server Manager* لإطفاء قاعدة معطيات أوراكل عن طريق تعليمة *SHUTDOWN* التي تأخذ الشكل التالي:

SHUTDOWN
[NORMAL]
or [IMMEDIATE]
or [TRANSACTIONAL]
or [ABORT]

١-إطفاء قاعدة معطيات ضمن الشروط العادية **Shutting Down a Database Under Normal Condition:**

شروط الإطفاء العادية تتضمن مايلي:

- عدم السماح بإجراء اتصالات جديدة بعد تنفيذ تعليمة الإطفاء.

- قبل إطفاء القاعدة، يقوم أوراكل بانتظار إلغاء الاتصال من قبل جميع المستخدمين الحاليين.
- لن يحتاج الإقلاع القادم إلى إجرائيات استرداد حالة instance recovery.



Shut Down Normal;

٢-إطفاء قاعدة معطيات بشكل فوري Shutting Down a Database Immediately:

تحتاج إلى عملية إطفاء فورية لقاعدة المعطيات في إحدى الحالات التالية فقط:

- حصول عملية إطفاء في التغذية الكهربائية في وقت قريب جدا.
- في حال عمل قاعدة المعطيات أو أحد تطبيقاتها بشكل غير نظامي.

هذه العملية تؤدي إلى حدوث الحالات التالية:

- إنهاء فوري لتعليمة SQL الحالية والمعالجة من قبل النظام.
- يتم التراجع عن أية تحويلات غير مثبتة Uncommitted Transaction.
- لا ينتظر النظام من المستخدمين الحاليين إلغاء الاتصال وإنما يقوم بنفسه بإجراء عملية إلغاء الاتصال هذه.
- ربما تحتاج عملية الإقلاع القادمة إلى قاعدة المعطيات لاسترداد الهيئة instance recovery (وهو ما يقوم به أوراكل تلقائيا).



Shutdown Immediate;

٣-إحباط هيئة Aborting an Instance:

يمكنك إطفاء قاعدة معطيات بشكل فوري بإحباط هيئة قاعدة المعطيات هذه.

حاول عدم استخدام هذه الطريقة إلا في الحالات التالية فقط:

- في حال عدم عمل قاعدة المعطيات أو أحد تطبيقاتها بشكل نظامي وفي حال لم تفلح أية طريقة من طرق الإطفاء السابقة.

- تحـــتاج إلى إطفـــاء القـــاعدة فـــوراً ربما بسبب وجود إنذار بانقطاع التيار الكهربائي بشكل فوري.
- حدوث مشاكل خطيرة عند إقلاع هيئة قاعدة المعطيات.

تؤدي هذه الطريقة إلى حدوث الحالات التالية:

- إنهاء فوري لتعليمة SQL الحالية والمعالجة من قبل النظام.
- لايتمّ التراجع عن التحويلات غير المثبتة.
- لا ينتظر النظام من المستخدمين الحاليين إلغاء الاتصال وإنما يقوم بعمل ذلك بشكل فوري.
- ستحتاج عملية الإقلاع القادمة للقاعدة إلى عملية استرداد هيئة.

من أجل القيام بإنهاء هيئة قاعدة معطيات استخدم تعليمة SHUT DOWN مع الخيار ABORT على الشكل:

SHUTDOWN ABORT;

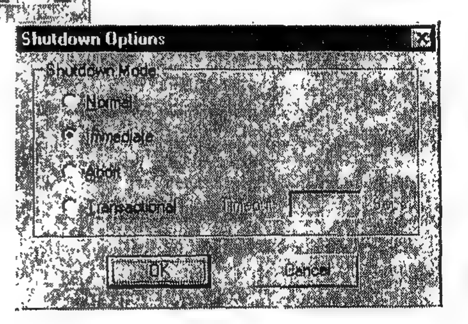
إطفاء قاعدة معطيات أوراكل من خلال Instance Manager

من الأداة Instance Manager، حـــدد الخيار Shutdown، كما في الشكل١٩-٤.



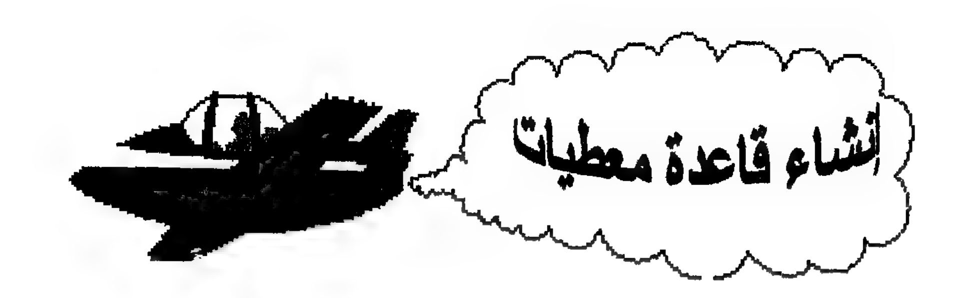
الشكل ١٩-٤

أنقـــر زر Apply، يظهـــر صندوق حوار يطلب تحديد نمط الإطفاء. أنظر الشكل ١٩ -٥.



الشكل ١٩-٥

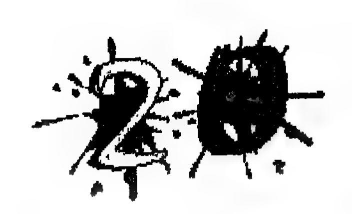

الباب السادس



٢٠ . إدارة قاعدة معطيات.

٢١ . إدارة الفضاءات الجدولية.

٢٢ . إدارة مقاطع التراجع.



إدارة قاعدة معطيات

Administrating a Database

إنّ إنشاء قاعدة معطيات ما يحتاج إلى تعليمة لغة تعريف معطيات DDL واحدة، لكن قد تحتاج إلى أسابيع وحتى أشهر من التحضير لإجراء هذه العملية.

من أجل إنشاء قاعدة معطيات، يجب عليك معرفة الكثير حول المعطيات التي سـيتم تخزيـنها في القاعدة وحجم هذه المعطيات. جميع المعطيات السابقة تستخدم لتحديد ملفات المعطيات وملفات الإرجاع وهي أحدى مهام مدير قاعدة المعطيات DBA.

ضمن نظام WINDOWS NT، يجب عليك إنشاء الهيئة قبل إنشاء قاعدة المعطيات لأن أوراكل يعمل كخدمة ضمن NT، لذلك فإن الهيئة ضرورية لمرحلة إنشاء القاعدة.

يـتم إنشـاء قاعدة معطيات في مرحلتين مرتبطتين، تتضمن المرحلة الأولى تعليمة إنشاء القاعدة وهي تؤدي إلى إنشاء ملفات الإرجاع وملفات التحكم وملفات المعطيات الضرورية لإنشاء الفضاء الجدولي SYSTEM.

وكما ذكرنا سابقاً يحتوي الفضاء الجدولي SYSTEM على مقطع التراجع SYSTEM وعلى قاموس المعطيات والإجرائيات المخزّنة إضافةً إلى البنى الأخرى الضرورية لتشغيل هيئة أوراكل.

أما المرحلة الثانية فهي تتضمن إضافة الفضاءات الجدولية والجداول والفهارس اللازمة لتخزين معطياتك الخاصة.

توجد مجموعة من العوامل الواجب اعتبارها عند تصميم قاعدة المعطيات أهمها:

١- **حجم قاعدة المعطيات:** يجب أن تكون قادراً على التعامل مع كمية المعطيات التي سيتم تخزينها في القاعدة، طبعاً بالنسبة لقواعد المعطيات الصغيرة فهذا العامل غير ضروري، أما بالنسبة لقواعد المعطيات الكبيرة فهو من أهم العوامل الواجب أخذها بعين الاعتبار. من أجل ذلك يجب أن تكون متأكداً بأنه ليس فقط لديك مساحة كافية لملفات المعطيات، وإنما أيضاً للفهارس المرتبطة بها. وفي بعض الحالات قد تحتاج إلى مساحة تخزين مؤقتة من أجل نسخ ملفات الدخل إليها قبل شحنها إلى قاعدة المعطيات.

توجد ضمن قاعدة المعطيات أوراكل بعض القيود الواجب معرفتها حول حجوم أجزاء القاعدة وهي:

* الحجم الأعظم لملفات المعطيات 32 G. B.
* العدد الأعظم لملفات المعطيات في كل فضاء جدولي هو ١٠٢٢ ملف.
* الحجم الأعظم للفضاء الجدولي هو 32 T. B.

٢- **الأداء Performance:** وهو عامل هام يجب مراعاته عند تصميم قاعدة المعطيات، لذلك يجب محاولة استغلال جميع إمكانيات التجهيزات المتوفرة لديك من أجل تحسين أداء القاعدة أكبر قدر ممكن.

مثلاً إذا احتوى جهازك على قرصين صلبين، يفضّل وضع جداول القاعدة على قرص والفهارس على القرص الآخر بحيث يتم تشغيلهما في آن واحد عند البحث عن معلومات ضمن القاعدة.

كذلك من الأفضل تقسيم قاعدة المعطيات إلى فضاءات جدولية مختلفة حسب عملها. يمكّنك هذا من إجراء عمليات الصيانة والنسخ الاحتياطي لكل فضاء جدولي بشكل منفصل عن الآخر .

٣- **حماية المعطيات Data Protection**: من أهم واجبات مدير قاعدة المعطيات حماية المعطيات في النظام وطريقة تنفيذ هذه الحماية.

كما سنرى لاحقاً، فإن كلّ تغيير يقوم به أوراكل على قاعدة المعطيات تتم كتابته على ملفات الإرجاع redo log files ومن ثم تتم أرشفة هذه الملفات.

هذه الملفات سيتم استخدامها فيما بعد عند إجراء عملية النسخ الاحتياطي كي تتمكن من استرداد قاعدة المعطيات إلى النقطة التي حدث فيها تعطل النظام.

لذلك فإنه من الضروري جداً حماية ملفات الإرجاع redo log files وملفات الإرجاع المؤرشفة archive log files من تعطّل وسائط التخزين.

٤- **التقسيم Partitioning**: توجد مع Oracle8 طريقة جديدة لتقسيم المعطيات تمكّنك من تقسيم الجداول التي تحتوي على معطيات مقسّمة إلى مجالات.

وعلى اعتبار أن التقسيم يتم على مستوى الفضاء الجدولي، والفضاءات الجدولية مؤلفة من ملفات معطيات، فإنه من الضروري تخطيط عملية التقسيم مثل بناء ملفات المعطيات.

التحضير لإنشاء قاعدة معطيات

عند إنشاء قاعدة معطيات، يقوم نظام أوراكل بتحضير مجموعة ملفات ضمن نظام التشغيل يمكنها العمل سوية كقاعدة معطيات أوراكل.

تتضمن عمليات إنشاء قاعدة معطيات مايلي:

- إنشاء ملفات معطيات data files جديدة أو استخدام معطيات موجودة ضمن ملفات معطيات قديمة.
- إنشاء بنى يحتاجها أوراكل للوصول إلى قاعدة المعطيات واستخدامها (قاموس المعطيات).
- إنشاء ملفات التحكم Control Files وملفات الإرجاع Red dog Files الخاصّة بقاعدة المعطيات.

يتم إنشاء قاعدة معطيات باستخدام تعليمة CREATE DATABASE الموجودة في لغة SQL، لكن يجب أولاً تحديد مايلي:

- تحديد جداول وفهارس القاعدة، وتوقع المساحة التخزينية المطلوبة.

* تحديد كيفيّة حماية القاعدة الجديدة.
* تحديد مجموعة محارف Character Set قاعدة المعطيات، لتعريف اللغة التي سيتم فيها تخزين المعطيات.

ولكي تستطيع إنشاء قاعدة معطيات جديدة يجب أن تمتلك العناصر التالية:

* امتيازات نظام التشغيل المرتبطة مع إمكانيات كاملة لإدارة قاعدة المعطيات (الاتصال بالحساب INTERNAL).
* ذاكرة كافية لتشغيل هيئة أوراكل ORACLE Instance.
* مساحة تخزين كافية على قرص التخزين من أجل تخطيط قاعدة المعطيات.

إنشاء قاعدة معطيات أوراكل جديدة

توجد مجموعة من الخطوات الواجب اتباعها عند قيامك بإنشاء قاعدة معطيات جديدة وهي:

١-الخطوة الأولى: إنشاء نسخ احتياطية لقواعد المعطيات Backup Existing Databases.

قبل قيامك بإنشاء قاعدة معطيات جديدة، فإنه ينصح بشدة أن تقوم بإنشاء نسخة احتياطية كاملة لجميع قواعد المعطيات الموجودة وذلك في حال حدوث خطأ ما يمكن أن يؤدي لتلف بعض الملفات الموجودة.

النسخ الاحتياطي يجب أن يتضمن ملفات الوسطاء Parameter files، ملفات المعطيات Data files، ملفات الإرجاع Redo log files، وملفات التحكم Control files.

٢-الخطوة الثانية: إنشاء ملفات الوسطاء Creating Parameter files.

يتم تشغيل هيئة قاعدة المعطيات instance database (SGA وإجراءات الخلفية background processes) باستخدام ملفات الوسطاء.

ويجب أن تحتوي كلّ قاعدة المعطيات في النظام على ملف وسطاء واحد على الأقل ويرتبط بهذه القاعدة فقط. ويجب عدم استخدام نفس ملف الوسطاء لعدة قواعد معطيات.

عند قيامك بإنشاء ملف وسطاء جديد، يمكنك نسخ ملف الوسطاء الموجود ضمن نسخة ORACLE لديك أعط هذه النسخة اسماً جديداً، تستطيع بعدها التعديل على هذا الملف وفق احتياجات القاعدة الجديدة.

٣-الخطوة الثالثة: تحرير ملفات الوسطاء الجديدة Edit New Parameter Files.

عندما تقوم بإنشاء قاعدة معطيات جديدة، حاول تحديد قيم الوسطاء التاليـــة (لمزيـد مـن التفاصيل راجع الملحق٣):

DB_NAME, DB_DOMAIN, CONTROL_FILES, DB_BLOCK_SIZE, DB_BLOCK_BUFFERS, PROCESSES, ROLLBACK_SEGMENTS.

يفضل أيضاً تحديد قيم الوسطاء التالية:

LICENSE_MAX_SESSION, LICENSE_SESSION_WARNING, LICENSE_MAX_USERS.

سنقوم فيما يلي بشرح كلّ من هذه الوسطاء:

- الوسيطان **DB_NAME, DB_DOMAIN**: بتحديد قيمتي هذين الوسيطين يتم إنشاء اسم قاعدة معطيات عام global database name (الاسم والموقع ضمن بنيـة الشبكة) وذلك قبل إنشاء القاعدة لأنه ليس من السهل بعد ذلك تعديل اسم القاعدة.

ويحدد الوسيط DB_NAME الاسم المحلي لقاعدة المعطيات.

أما الوسيط DB_DOMAIN فيحدد مجاله (منطقياً) ضمن بنية الشبكة.

لنفترض أننا نريد إنشاء قاعدة معطيات باسم عام TEST.SYR.COM يتم ذلك بتحديد قيم هذين الوسيطين على الشكل:

DB_NAME=TEST
DB_DOMAIN=SYR.COM

- الوسيط **CONTROL_FILES**: لتحديد أسماء ملفات التحكم لقاعدة المعطيـات الجديدة ويجب ألا يكون أي من هذه الملفات موجوداً من قبل (لأنه ستتم الكتابة فـوق الملف القديم).

وننصحك بشدة باستخدام ملفي تحكم على الأقل وذلك على سواقتي أقراص منفصلتين.

- الوسيط **DB_BLOCK_SIZE**: لتحديد حجم كتلة المعطيات. ويكون عـادة 2k أو 4k ويفضل أخذ القيمة الافتراضية لحجم كتلة معطيات نظام التشـغيل، ولا يمكـن تعديل قيمة هذا الوسيط بعد إنشاء القاعدة.

- الوسيط **DB_BLOCK_ BUFFERS**: لتحديد عدد الذواكر المؤقتـة buffer وذلك في منطقة النظام العامة System Global Area (SGA). ويؤثر هذا العـدد على أداء الذاكرة المخبئية Cache، فإذا كان حجم الذاكرة المخبئية كبيراً فهو يقلـص

عدد عمليات الكتابة على القرص عند تعديل المعطيات، لكنه قد يؤثر على الذاكرة العامة.

- الوسيط PROCESS: يحدد هذا الوسيط العدد الأعظم لإجراءات نظام التشغيل التي يمكنها الاتصال مع أوراكل بشكل متزامن.

يجب أن يتضمن هذا الوسيط 5 إجراءات مخصصة للخلفية background processes وإجراء واحد لكل مستخدم

مثلاً إذا كان لدينا 50 مستخدم، يجب تحديد قيمة هذا الوسيط بـ 55 على الأقل.

- الوسيط ROLLBACK-SEGMENTS: يحدد هذا الوسيط قائمة مقاطع التراجع التي تحتاجها قاعدة المعطيات للإقلاع.

يحدد الجدول التالي كيفية تحديد عدد مقاطع التراجع التي تحتاجها:

عدد التحويلات المتزامنة	عدد مقاطع التراجع الموافقة
N<16	4
16<=n<32	8
N<=n	N/4 ولا تتجاوز ٥٠

بعد تركيب نسخة ORACLE، يجب أن تقوم بإنشاء مقطع تراجع واحد على الأقل ضمن الفضاء الجدولي SYSTEM قبل أن تستطيع إنشاء أي مخطط عناصر Schema objects جديد.

- الوسيط LICENCE_MAX_SESSIONS: لتحديد العدد الأعظم للدورات المتزامنة التي يمكنها الاتصال مع قاعدة المعطيات الموجودة على حاسب محدد.
- الوسيط LICENSE_SESSION_WARNING: لتحديد العدد الأعظم للمستخدمين الذين يتم إنشاؤهم في قاعدة المعطيات.

٤-الخطوة الرابعة: التحقق من محدد الهيئة Instance Identifier

وذلك في حال وجود قواعد معطيات أخرى ضمن النظام، هذه الإجرائية خاصة بنظام التشغيل.

٥-الخطوة الخامسة: تشغيل برنامج Server Manager والاتصال بالحساب INTERNAL.

٦-الخطوة السادسة: تشغيل الهيئة Instance.

في هذه المرحلة لا توجـــد قـــاعدة معطيـــات وإنمـــا فقـــط SGA وإجـــراءات الخلفيـــة background processes تكون فعالة وتتحضر لإنشاء قاعدة المعطيات الجديدة.

STARTUP NOMOUNT PFILE= path\ filename

٧-الخطوة السابعة: إنشاء قاعدة المعطيات Create The Database

من أجل إنشاء قاعدة معطيات جديدة استخدم تعليمة CREATE DATABASE وهـــي تمكّنك من إجراء العمليات التالية:

- إنشاء ملفات معطيات data files لقاعدة المعطيات.
- إنشاء ملفات التحكم control files لقاعدة المعطيات.
- إنشاء ملفات الإرجاع redo log files لقاعدة المعطيات.
- إنشاء الفضاء الجدولي SYSTEM ومقاطع التراجع SYSTEM .
- إنشاء قاموس المعطيات Data Dictionary.
- إنشاء المستخدمين SYS و SYSTEM.
- تحديد مجموعة المحارف Character Set المستخدمة لتخزين المعطيـــات فـــي القاعدة.
- تركيب mount وفتح قاعدة المعطيات للاستخدام.

٨-الخطوة الثامنة: النسخ الاحتياطي لقاعدة المعطيات Backup The Database

وذلك للتأكد من حماية كامل ملفاتك في حال حدوث عطل ما.



في حال الفشل في إنشاء قاعدة المعطيات لأي سبب، قم بالخطوات التالية قبل محـــاولتك إعادة إنشائها من جديد:



١ً - أطفئ الهيئة Shutdown instance.

٢ً - احـــذف جميـــع الملفـــات التـــي تـــم إنشـــاؤها باســـتخدام تعليمـــة CREATE DATABASE.

بعد ذلك قم بإصلاح الخطأ الذي سبب فشل إنشاء القاعدة واستمر بدءاً من الخطوة السادسة.

الآن وبعد إنشاء قاعدة المعطيات، تبقى الهيئة فعالة وقاعدة المعطيـــات مفتوحـــة ومتاحـــة للاستخدامات العادية لقاعدة المعطيات. وتحتوي هذه القاعدة الجديدة على مستخدمين فقـــط هما SYS و SYSTEM كما ذكرنا سابقاً.

إنشاء قاعدة معطيات باستخدام Server Manager

يمكن إنشاء قاعدة معطيات باستخدام تعليمة OREATE DATABASE على الشكل:

CREATE DATABASE [[database]
[CONTROLFILE REUSE]]
LOGFILE [GROUP group-number] log file
[, [GROUP group-number] log file]...
[MAXLOGFILES number]
[MAXLOGMEMBERS number]
[MAXLOGHISTORY number]
[MAXDATAFILES number]
[MAXINSTANCES number]
[ARCHIVELOG or NORCHIVELOG]
[EXCLUSIVE]
[CHARACTER SET charset]
[DATAFILE fil_specification [AUTOEXTEND off \ON
[NEXT number KIM]
[MAXSIZE UNLIMITED \ number KIM]
[, DATAFILE file_specification [AUTOEXTEND OFF \ON
[NEXT number KIM]
[MAXSIZE UNLIMITED \ number KIM]]

حيث:

- **Database**: اسم قاعدة المعطيات التي سيتم إنشاؤها، ويمكن أن يصل إلى ثمانية أحرف.
- **CONTROLFILE REUSE**: يحدد هذا الوسيط الاختياري بأنه يمكن إعادة الكتابة على أي ملف تحكم موجود مسبقاً. وفي حال عدم اختيار هذا الوسيط فإنه تعليمة CREATE DATABASE ستفشل إذا وجد ملف التحكم مسبقاً.
- **LOGFILE**: لتحديد اسم ملف الإرجاع، ويمكن أيضاً بشكل اختياري تحديد مجموعة الإرجاع مع الوسيط الاختياري GROUP.
- **MAXLOGFILES**: لتحديد العدد الأعظم لمجموعات ملفات الإرجاع التي يمكن إنشاؤها لهذه القاعدة.
- **MAXLOGMEMBERS**: لتحديد العدد الأعظم لأعضاء ملفات الإرجاع في مجموعة ملفات الإرجاع.

- **MAXLOGHISTORY**: يستخدم هذا الوسيط مع المخدّم المتوازي وهو يحدد العدد الأعظم من ملفات الإرجاع المؤرشفة التي ستستخدم في عمليات الاسترداد recovery.
- **MAXDATAFILES**: يحدد هذا الوسيط العدد الأعظم من الملفات التي يمكن إضافتها إلى قاعدة المعطيات قبل أن يتم توسيع ملف التحكم بشكل تلقائي.
- **MAXINSTANCES**: لتحديد العدد الأعظم من الهيئات التي يمكن لقاعدة المعطيات فتحها بشكل متزامن.
- **ARCHIVELOG**: لتشغيل قاعدة المعطيات بنمط ARCHIVELOG حيث تتم في هذا النمط أرشفة مجموعة ملفات الإرجاع قبل إعادة استخدامها وهذا النمط ضروري عند محاولة استرداد قاعدة المعطيات recovery.
- **NOARCHIVELOG**: لتشغيل قاعدة المعطيات بنمط NOARCHIVELOG حيث لا تتم أرشفة مجموعات ملفات الإرجاع. وهو الخيار الافتراضي.
- **EXCLUSIVE**: يحدّد هذا الوسيط تركيب قاعدة المعطيات وفق نمط EXCLUSIVE بعد إنشاء هذه القاعدة. في هذا النمط يمكن لهيئة وحيدة فقط تركيب القاعدة.
- **CHARACTER SET**: لتحديد مجموعة المحارف التي سيتم تخزين المعطيات وفقها.
- **NATIONAL CHARACTER SET**: لتحديد مجموعة المحارف العالمية المستخدمة لتخزين المعطيات في الأعمدة NVARCHAR2, NCLOB, NCHAR.
- **DATAFILE**: يحدد هذا الوسيط الملف المعرف بالاسم filename وحجم هذا الملف SIZE بالكيلوبايت K أو الميغابايت M.

كما يمكن تفعيل خيار التوسيع التلقائي AUTOEXTEND ON أو إلغاء تفعيله AUTOEXTEND.

وفي حال تفعيل خيار التوسيع التلقائي يمكن تحديد العدد بالكيلوبايت K أو الميغابايت M الذي ستتم إضافته بشكل تلقائي إلى ملف المعطيات الذي سيتم توسيعه تلقائياً في كل مرة.

ويمكن تحديد خيار توسيع ملف المعطيات بشكل غير محدد MAXSIZE UNLIMITED أو بشكل محدد:

MAXSIZE number (K or M)



```
CREATE DATABASE test CONTROLFILE REOSE
LOGFILE
GROUP 1 ('C:\ TEST_DB\ log 1 a. dbf';
            'd: \ TEST_DB\ log 1 b. dbf') SIZE 100 K,
GROUP 2 ('C: \TEST_DB\ log 2 a. dbf;
            'd: \TEST_DB\log2 b. dbf') SIZE 100  K
DATAFILE 'C: \TEST_DB\ data 1. Dbf' SIZE 10 M,
            'C: \ TEST_DB\ data 2 . dbf ' SIZE 10 M
AUTOEXTEND ON NEXT 10 M MAXSIZE 50 M;
```



إنشاء قاموس المعطيات Creating Data Dictionary

بعد إنشاء قاعدة معطيات، يجب تنفيذ الملفين CATALOG.SQL و CATPROC.SQL لبـــناء مشـــاهد قاموس المعطيات وذلك في حال بناء القاعدة بشكل يدوي باستخدام الأداة Server Manager، أمـــا عـــندما نقوم باستخدام الأداة Oracle Database Assistant فيقوم تلقائياً بتوليد قاموس المعطيات.

هذه المشاهد ضرورية للعمليات الخاصة بالنظام ولمدير قاعدة المعطيات.

حيث يتم تنفيذها على الشكل:

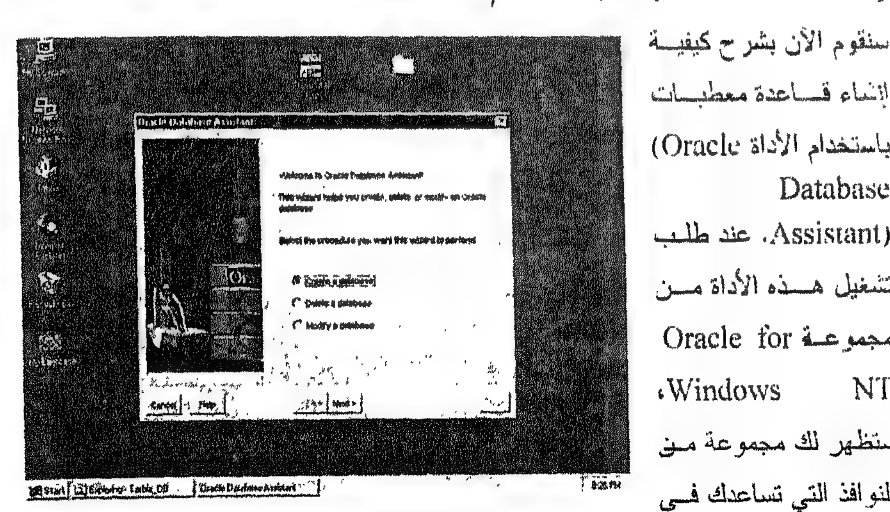
@ C: \ ORANT \ RDBMS 80 \ ADMIN \ CATALOG;
@ C: \ ORANT \ RDBMS 80 \ ADMIN \ CATPROC;

(بافتراض أن نظام أوراكل مثبت على السواقة :C).

انظـــر المـــلحق٢ الـــذي يوضـــح التعليمات الموافقة للمثال الذي تمّ توليده باستخدام الأداة Database Assistant لإنشـــاء قـــاعدة المعطيات TEST والفضاءات الجدولية ومقاطع التراجع الموافقة.



إنشاء قاعدة معطيات باستخدام Oracle Database Assistant

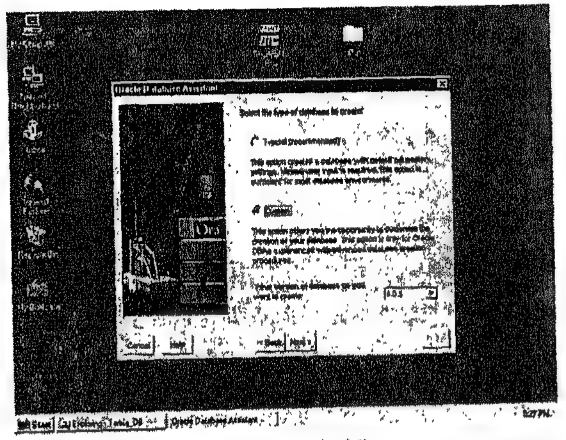


الشكل ٢٠-١

سنقوم الآن بشرح كيفية إنشاء قاعدة معطيات باستخدام الأداة (Oracle Database Assistant). عند طلب تشغيل هذه الأداة من مجموعة Oracle for Windows NT، ستظهر لك مجموعة من النوافذ التي تساعدك في إنشاء القاعدة:

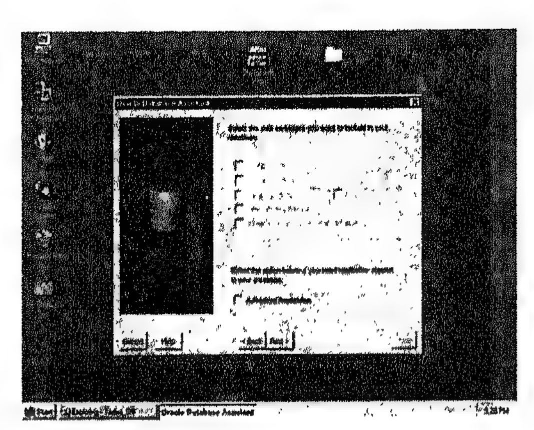
١. النافذة الأولى تطلب اختيار العملية المطلوب تنفيذها إما إنشاء قاعدة معطيات Create a database، أو حذف معطيات قاعدة Delete a database، أو تعديل قاعدة معطيات Modify a database. سنختار الآن خيار إنشاء قاعدة المعطيات (أنظر الشكل ٢٠-١).

٢. النافذة الثانية، تطلب اختيار نمط إنشاء قاعدة المعطيات (انظر الشكل٢٠-٢)، رأينا من قبل طريقة Typical والآن سنختار Custom.



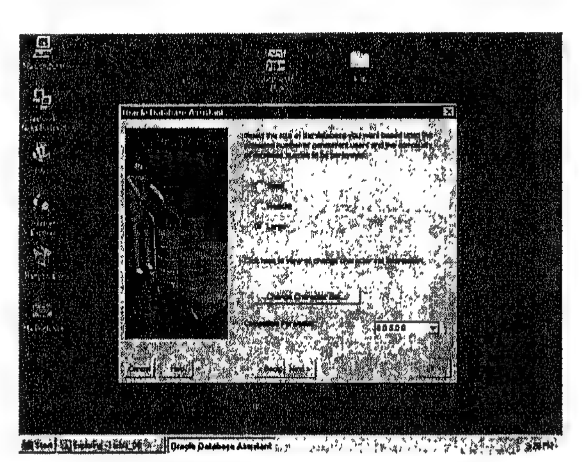
الشكل ٢٠-٢

٣. النافذة الثالثة، وفيها الطلب هل ترغب باختيار تناسخ القاعدة Replication، حيث ستتم إضافة مستخدمين افتراضيين وحزم برمجية مخزنة، انظر الشكل ٢٠-٣:



الشكل ٢٠-٣

٤. في النافذة الرابعة، تستطيع تحديد حجم قاعدة المعطيات، كذلك مجموعة المحارف Character set ووسيط التوافق Compatibility Parameter. انظر الشكل ٢٠-٤.



الشكل ٢٠-٤

٥. أنقر على زر Change Character Set لتحديد مجموعة المحارف التي سيتم التعامل معها في قاعدة المعطيات. للتعامل مع معطيات باللغة العربية حدّد قيم مجموعة المحارف كما في الشكل ٢٠-٥.

Character Sets

View or change the character set(s) for your database

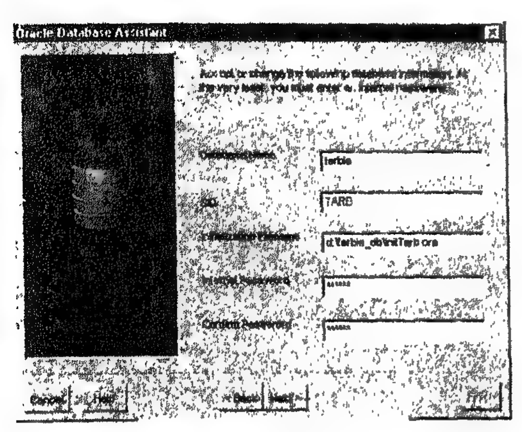
Character Set: AR8ISO8859P6

National Character Set: AR8ISO8859P6

OK | Cancel

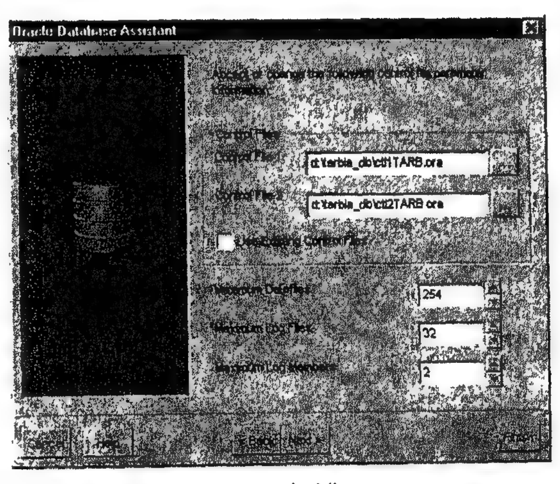
الشكل ٢٠-٥

٦. في النافذة السادسة، يمكنك تحديد اسم قاعدة المعطيات data base name ومحدد النظام SID وملف الوسطاء الابتدائي Initialization filename وكلمة مرور حساب Internal، Internal Password انظر الشكل ٢٠-٦.



الشكل ٢٠-٦

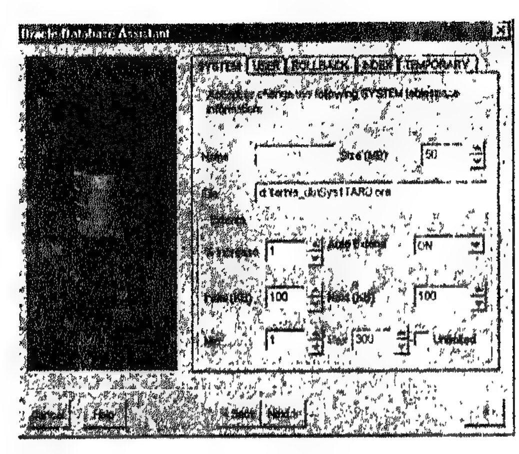
٧. في النافذة السابعة، يمكنك تحديد ملفات التحكم Control Files والعدد الأعظمي لملفات المعطيات Maximum Data files والعدد الأعظمي لملفات الإرجاع Maximum log files والعدد الأعظمي لأعضاء الإرجاع Maximum log Members. انظر الشكل ٢٠-٧.



الشكل ٢٠-٧

٨. أما في النافذة الثامنة، فسترى، خمسة أبواب تسمح لك بتحديد حجوم وخصائص ملفات الفضاءات الجدولية التالية:

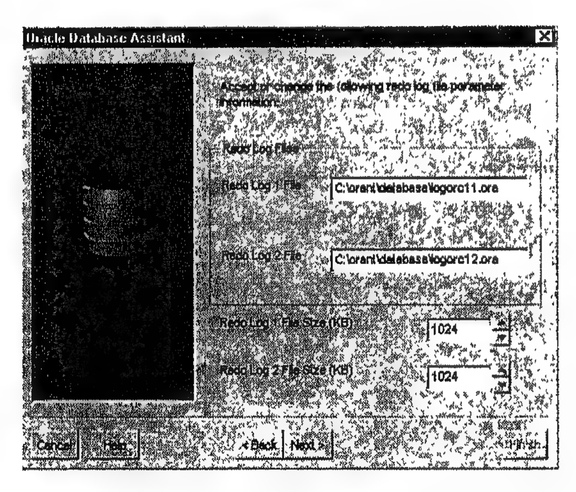
SYSTEM
USER
ROLLBACK
INDEXE
TEMPORARY



الشكل ٢٠-٨

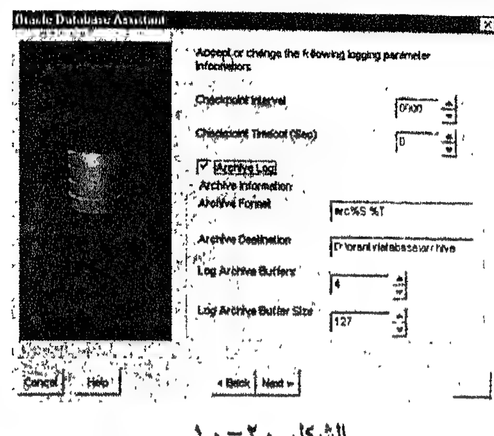
انظر الشكل ٢٠-٨. (لمزيد من التفاصيل حول الفضاءات الجدولية راجع الفصل ٢١).

٩. أما النافذة التاسعة فتسمح لك بتحديد أسماء ومواقع ملفات الإرجاع Log Files وأحجام هذه الملفات. انظر الشكل ٢٠-٩.



الشكل ٢٠-٩

١٠- أنقر زر Next تظهر نافذة جديدة تمكّنك من تحديد مجال نقطة التحقّق Checkpoint interval وزمن انقضاء نقطة التحقّق Checkpoint Timeout، كذلك تحديد معلومات أرشفة ملفات الإرجاع Archive Information في حال قمت بتفعيل صندوق التحقق Archive Log. انظر الشكل ٢٠-١٠. (لمزيد من المعلومات أنظر الفصل ٢٥).

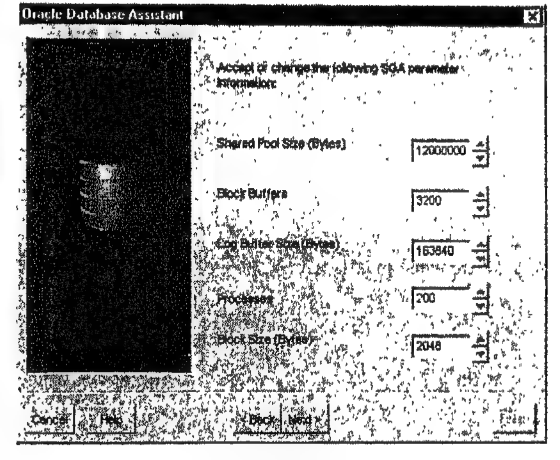


الشكل ٢٠-١٠

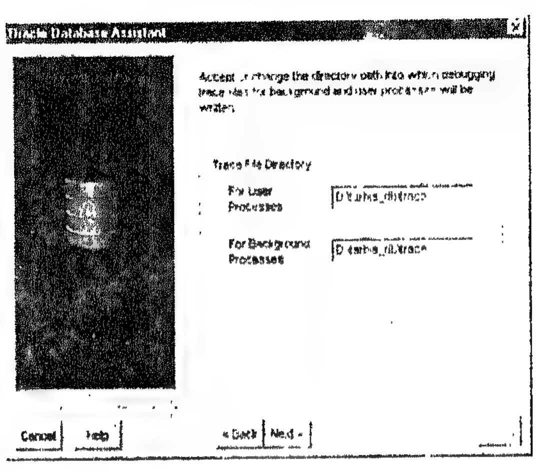
١١- النافذة التالية تساعدك في تحديد الوسطاء الابتدائية التي تؤثر على ذاكرة النظام العامة SGA وهي:

Shared pool size (SHARED_ POOL_SIZE)
Block Buffers (DB_ BLOCK_ BUFFERS)
Log Buffer (LOG_ BUFFER)
Processes (DB_ BLOCK_SIZE)

انظر الشكل ٢٠-١١.

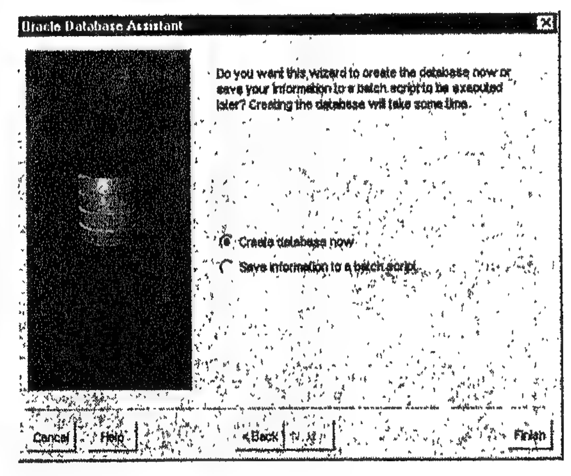


الشكل ٢٠-١١



الشكل ٢٠-١٢

١٢- أنقر زر Next، تظـهر نافذة جديدة تمكنك من تحديـد موقع مجلد ملفات الأثر Trace الخاصة بإجراءات المستخدمين وإجـراءات الخلفيـة. انظـر الشكل ٢٠-١٢.



الشكل ٢٠-١٣

١٣- النافذة الأخيرة تسـألك إن كنت راغباً بالبدء بإنشاء قـاعدة المعطيات Create database now أو ترغب بحفظ معلومات الإنشاء في ملف دفعـي Save information to a batch script (يمكنـك مراجعـة الملحق٢ للتعرف على الملـف الدفعي الذي يتمّ توليده). انظـر الشكل ٢٠-١٣.

ابدأ بإنشاء قاعدة المعطيات وانقر زر Finish.

وعندما تنقر زر Yes تظهر نافذة تعطيك معلومات عن القاعدة التي سيتم إنشاؤها كاسـم القاعدة ومحدد النظام SID. انقر زر OK وانتظر مدة من الوقت قبل إنشاء القاعدة.

Modifying the Database تعديل قاعدة معطيات

تعديل قاعدة معطيات باستخدام *Server Manager*

يمكننا القيام بذلك من خلال تعليمة ALTER DATABASE ضمن Server Manager والتي تأخذ الشكل العام:

ALTER DATABASE [database]
[MOUNT [STANDBY DATABASE] [EXCLUSIVE \ PARALLEL]]
[CONVERT]
[OPEN [RESETLOGS \ NORESETLOGS]]
[ACTIVATE STANDBY DATABASE]
[ARCHIVELOG \ NDARCHIVELOG]
[RECOVER recover_parameters]
[ADD LOGFILE [THREAD number] [GROUP number] logfile
[, [GROUP number] logfile]...]
[ADD LOGFILE MEMBER 'filename [REUSE]
[,'filename' [REUSE]
[, 'filename' [REUSE]...] [TO GROUP number] or
[, 'filename' [,'filename']...]
[,'filename' [REUSE] [,'filename' [REUSE]...
[TO GROUP number] or ['filename' [,'filename']...]]
[DROP LOGFILE [GROUP number] or ['filename' [,'filename']...]
[, GROUP number] or ['filename'[,'filename']...]]
[DROP LOGFILE MEMBER 'filename' [, 'filename']...]
[CLEAR [UNARCHIVED] LOGFILE
[GROUP number] or ['filename' [, 'filename'] ...]
[UNRECOVERABLE DATAFILE]
[RENAME FILE 'filename' [,'filename']... to 'filename'
[,'filename']...
[CREATE STANDBY CONTROLFILE AS 'control_file_name'
[REUSE]]
[BACKUP CONTROLFILE
[TO'filename'[REUSE]] or [TO TRACE [RESETLOGS or
NORESELOGS]]
[RENAME GLOBAL NAME to database [, domain]...]
[RESET COMPATIBILITY]

[SET [DBLOW= value] or [DBHIGH= value] or [DBMAC ON or OFF]]
[ENABLE [PUBLIC] THREAD number]
[DISABLE THREAD number]
[CREATE DATAFILE 'filename' [,'filename']...
AS filespec [,'filespec]...]
DATAFILE 'filename' [,'filename']...
ONLINE or OFFLINE [DROP] or RESIZE number (K or M)
Or AUTOEXTEND OFF or ON
[NEXT number (K or M)]
[MAXSIZE UNLIMITED or number (K or M)]
or END BACKUP]

حيث:

☆ **Database**: اسم قاعدة المعطيات.

☆ **MOUNT**: لتركيب قاعدة المعطيات حيث تأخذ الخيارات:

- **MOUNT STANDBY DATABASE**: لتركيب قاعدة معطيات مستعدّة للعمل.
- **MOUNT EXCLUSIVE**: لتركيب قاعدة المعطيات بنمط EXCLUSIVE حيث يمكن في هذا النمط لهيئة وحيد فقط تركيب قاعدة المعطيات.
- **MOUNT PARALLEL**: لتركيب قاعدة المعطيات بنمط التوازي.

☆ **CONVERT**: لتحويل قاموس معطيات ORACLE 7 إلى قاموس معطيات ORACLE8.

☆ **OPEN**: لفتح قاعدة المعطيات في الاستخدام العادي ويمكن استخدام الخيارين:

- **RESETLOGES**: مع هذا الخيار يتم إلغاء المعلومات الموجودة في ملف الإرجاع.
- **NORESETLOGS**: وهو الخيار المعاكس للسابق.

☆ **ACTIVATE STANDBY DATABASE**: لتحويل قاعدة المعطيات المستعدة للعمل Standby Database إلى قاعدة المعطيات الحالية الفعالة.

☆ ARCHIVELOG: لتشغيل قاعدة المعطيات بنمط NOARCHIVELOG وهذا النمط خطير جداً فقد لا تستطيع استرداد قاعدة المعطيات في حال حدوث عطل ما في أقراص التخزين.

☆ RECOVER: وهو خيار استرداد قاعدة المعطيات، وهو يأخذ الشكل:

RECOVER [AUTOMATIC] [FROM 'Path']
[[STANDBY] DATABASE]
[UNTIL CANCEL] or [UNTILE TIME 'time']
or [UNTIL CHANGE number]
or [USING BACKUP CONTROLFILE]...]
[TABLESPACE tablespace [, tablespace]...]
[DATAFILE 'filename' [,'filename']...]
[LOGFILE 'filename']
[CONTINUE [DEFAULT]]
[CANCEL]
[PARALLEL parallel_definition]

☆ **ADD LOGFILE 'logfile'**: يستخدم هذا الخيار لإضافة ملف إرجاع بالاسم 'logfile'. ويمكن بتحديد الخيار THREAD إضافة ملف الإرجاع هذا إلى إجراء مخدّم متوازي محدّد، وإلا فإنه سيضاف إلى الهيئة الحالية.

☆ **ADD LOGFILE MEMBER 'filename'**: لإضافة مجموعة بالاسم 'filename' إلى مجموعات ملفات الإرجاع الموجودة مسبقاً. ويجب تحديد الوسيط REUSE إذا كان الملف 'filename' موجوداً مسبقاً.

وتستخدم عدة خيارات هنا هي:

- **TO GROUP number**: يستخدم هذا الخيار إذا كنت تعرف رقم مجموعة ملفات الإرجاع.
- **TO GROUP 'filename'**: يستخدم هذا الخيار إذا كنت تعرف اسم مجموعة ملفات الإرجاع.
- **DROP LOGFILE**: يقوم هذا الخيار بحذف جميع أعضاء مجموعة ملفات الإرجاع. حيث يمكن تحديد رقم المجموعة GROUP number أو اسم المجموعة 'filename'.
- **DROP LOGFILE MEMBER 'filename'**: يستخدم هذا الخيار لحذف عضو أو أعضاء من مجموعة ملفات الإرجاع.

- ◻ CLEAR LOGFILE: يستخدم هذا الخيار لحذف وإعادة إنشاء ملف إرجاع. ويستخدم في حال حدوث مشكلة في ملف إرجاع موجود. وباستخدام الخيار UNARCHIVED يتم مسح محتوى ملف الإرجاع دون أرشفته. ويمكن تحديد مجموعة ملفات الإرجاع من خلال تحديد الخيار: GROUP number في حال معرفة واصف المجموعة أو من خلال تحديد اسم ملف الإرجاع 'filename'. أما الخيار UNRECOVERABLE DATAFILE فيستخدم إذا احتوى الفضاء الجدولي على ملف معطيات غير فعال offline.

☆ **RENAME FILE 'filename' to 'filename'**: يستخدم هذا الخيار لإعادة تسمية ملف معطيات أو ملف إرجاع ضمن ملف التحكم وليس في قرص التخزين.

☆ **CREATE STANDBY CONTROLFILE AS 'control_file_name'**: يستخدم هذا الخيار لإنشاء ملف تحكم مستعدّ للعمل standby بالاسم control_file_name ويستخدم الخيار REUSE لتحديد اسم ملف موجود لإعادة استخدامه.

☆ **BACKUP CONTROLFILE**: يستخدم هذا الخيار لإنشاء نسخة احتياطية من ملف التحكم وذلك إما بتحديد اسم الملف الذي سيتم نسخه إليه To 'filename' أو بكتابة تعليمات SQL إلى ملف الأثر الذي يمكن استخدامه لإعادة إنشاء ملفات التحكم to TRACE.

☆ **RENAME GLOBAL NAME TO**: يسمح لك هذا الخيار بتغيير اسم قاعدة المعطيات أو اسم المجال أو كليهما.

☆ **RESET COMPATIBILITY**: يسمح هذا الخيار بإجراء التوافق مع نسخة قديمة من أوراكل وذلك بعد تشغيل الهيئة.

☆ **SET**: تستخدم لتحديد قيم وسطاء نسخة trusted Oracle.

☆ **ENABLE [PUBLIC] THREAD number**: أمر خاص بالمخدّم المتوازي، ويستخدم لتفعيل تشعّب محدد برقم والخاص بمجموعات ملفات الإرجاع. وبإضافة الخيار PUBLIC يصبح بالإمكان استخدام هذا الإجراء من قبل أي هيئة أوراكل.

☆ DISABLE THREAD number: يسمح هذا الخيار بإلغاء تفعيل تتابع محدد برقم والخاص بمجموعات سلفات الإرجاع.

☆ CREATA DATAFILE 'filename': يسمح هذا الخيار بإنشاء ملف معطيات فقد بسبب حدوث عطل ما ولم تنشأ نسخة احتياطية منه.

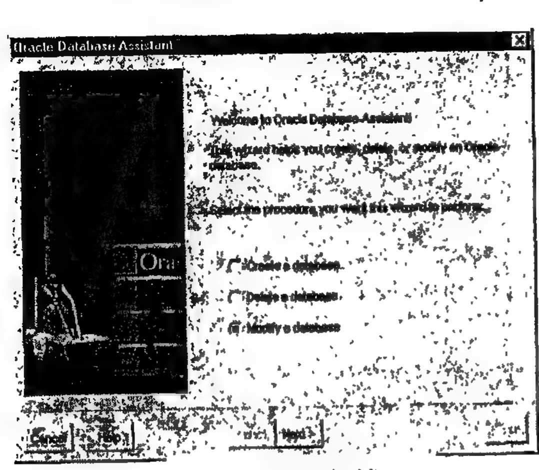
☆ AS filespec: يستخدم لتحديد وسطاء التوصيف.

☆ DATAFILE 'filename': يسمح لك هذا الخيار بتعديل حالة ملفات معطيات القاعدة إلى ONLINE أو OFFLINE أو بتغيير حجوم هذه الملفات RESIZE number (K or M) أو بتغيير حالة التوسع التلقائي AUTOEXTEND ON or OFF أو بتحديد تزايد هذه الملفات إما بشكل محدد: NEXT number (K or M) كذلك بتحديد الحجم الأعظم لملفات المعطيات MAXSIZE number (K or M) أو بعدم تقييد هذا الحجم: MAXSIZE UNLIMITED.

☆ END BACKUP: وذلك من أجل عدم إجراء عملية استرداد الوسط media recovery عند مقاطعة النسخ الاحتياطي بسبب تعطل هيئة أوراكل.

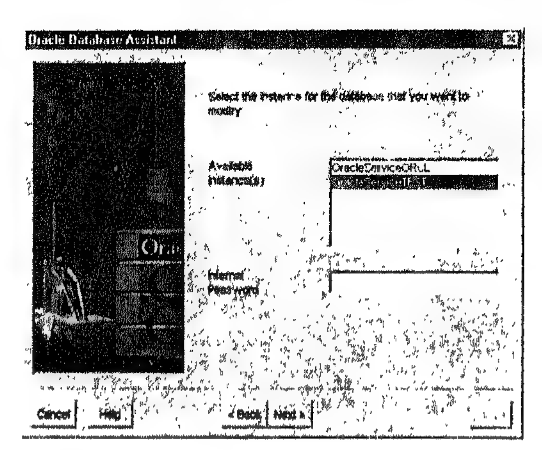
تعديل قاعدة معطيات باستخدام *Oracle Database Assistant*

في النسخة الجديدة من أوراكل Oracle8 Release 8.0.5.0.0 for Windows NT أصبح بالإمكان إجراء بعض عمليات التعديل على قاعدة المعطيات باستخدام الأداة Oracle Database Assistant، أنظر الشكل ٢٠-١٤.



الشكل ٢٠-١٤

انقر زر Next، تظهر نافذة جديدة تحتوي على قائمة بالهيئات المتاحة (أنظر الشكل ٢٠-١٥)، اختر الهيئة التي ترغب بتعديلها ثم أدخل كلمة مرور حساب Internal.

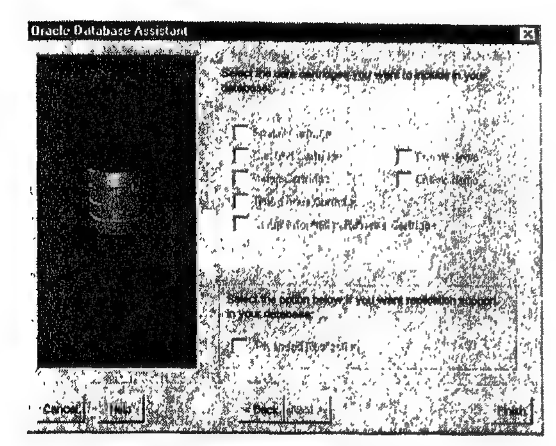


الشكل ٢٠-١٥

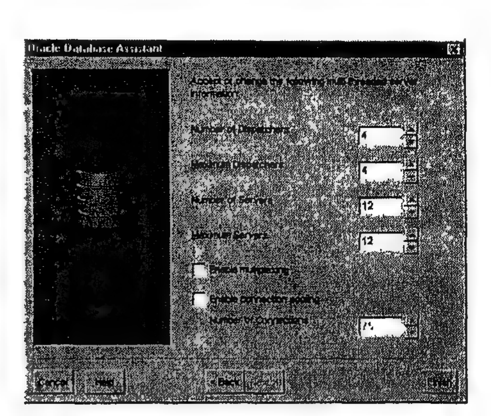
أنقر زر Next، تظهر نافذة جديدة تطلب طريقة تعديل قاعدة المعطيات، حيث يسمح لنا فقط بتعديل توصيف خرطوشة أوراكل Configure Oracle cartridge، أو بتأهيل خيار مخدّم أوراكل متعدّد التشعّبات enable multi-threaded server support (أنظر الشكل ٢٠-١٦).

الشكل ٢٠-١٦

في حال تحديد الخيار الأول تظـــهر نافذة تشبه الشكل ٢٠-١٧، تمكنّـــك مــن تحديـــد خرطوشـــة أوراكـــل المطلوب تأهيلها، أو اختيـــار دعـــم تناسخ المعطيات.

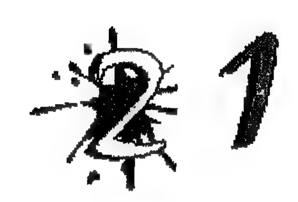


الشكل ٢٠-١٧



الشكل ٢٠-١٨

أما في حال تحديد الخيار الثاني، فتظـــهر نافذة تشبه الشكل ٢٠-١٨ تمكنـــك مـــن تحديد وسطاء المخدّم متعدّد التشعبات.



إدارة الفضاءات الجدولية

Administrating Tablespaces

بعد إنشاء قاعدة معطيات يجب الانتقال إلى الخطوة التالية من خطوات بناء التطبيقات وهـي إنشاء الفضاءات الجدولية، سنقوم أولاً بشرح وسطاء التخزين التي سيتم استخدامها لإنشاء الفضاءات الجدولية، ثم سنقوم بعد ذلك بشرح الطرق المستخدمة لإجراء العمليات الأساسية على الفضاءات الجدولية.

تحديد وسطاء التخزين Storage Parameters

يمكنك تحديد قيم وسطاء التخزين للأنماط التالية من بنى التخزين المنطقية :

- ☆ الفضاءات الجدولية Tablespaces.
- ☆ شرائح المعطيات (مثل: Snapshot log, Snapshots, Clusters, Table)
- ☆ شرائح الفهرسة Index segments.
- ☆ مقاطع التراجع Rollback segments.

يمكننا تحديد معنى كل وسيط من هذه الوسطاء والقيم التي يمكن أن يأخذها على الشكل التالي :

١- **INITIAL**: وهو الحجم (بالبايت) لأول مدى يتم حجزه عندما يتم إنشاء مقطع.

Default = 5 data block.
Minimum = 2 data block.
Maximum = operating system specific.

على الرغم من أن القيمة الإفتراضية محددة بالكتل blocks فإنه يفضل تحديد القيم بالبايت ويمكن استخدام الرموز M, K للدلالة على الكيلوبايت والميغابايت.

وتحديد القيم بالكتل متعلق بالحجم الافتراضي للكتلة والمحدد بالوسيط DB_BLOCK_SIZE.

فمثلاً إذا كان حجم كتلة المعطيات مساوٍ لـ ٢٠٤٨ byte فإن القيمة الافتراضية لهذا الوسيط تساوي ١٠٢٤٠B.

٢- **NEXT**: الحجم بالبايت للمدى التالي الذي سيتم إنشاؤه في الشريحة.

المدى الثاني سيأخذ الحجم الأصلي المحدد ضمن NEXT، وكل مدى جديد سيكون حجمه مساوٍ لحجم المدى السابق مضروباً بـ (١+ PCTINCREASE /١٠٠)

Default: 5 data blocks
Minimum: 1 data block
Maximum: Operating System Specific

٣- **MAXEXTENTS**: العدد الأعظم لكتل المدى.

Default: حسب حجم الكتلة ونظام التشغيل
Minimum: 1 (extent)
Maximum: Operating System Specific

٤- **MINEXTENTS**: العدد الأصغر للمدى التي سيتم إنشاؤها عند إنشاء مقطع مما يسمح بحجز مساحة كبيرة أثناء الإنشاء حتى لو لم يكن هناك مساحات فارغة متجاورة.

Default: 1 (extent)
Minimum: 1 (extent)
Maximum: Operating System Specific

٥- **PCTINCREASE**: النسبة المئوية لتوسيع حجوم المدى التي سيتم إنشاؤها.

Default: 50 (%)
Minimum: 0 (%)
Maximum: Operating System Specific

وباستخدام هذا المعامل بشكل صحيح، يمكننا تقليص تناثر المقاطع بتوسيع المدى الجديد وتقليل عدد المدى التي سيتم إنشاؤها في المقطع.

٦- **INITRANS**: لحجز مساحة ابتدائية لعدد مرات الوصول إلى كتل المعطيات بشكل متزامن، هذه المساحة تكون محجوزة في ترويسات جميع كتل المعطيات المرتبطة بمقاطع المعطيات أو الفهارس.

القيمة الافتراضية هي 1 للجداول ، و2 للفهارس والتجمعات.

٧- **MAXTRANS**: تحدد عدد مداخل التحويلات Transaction entries التي يمكنها بشكل متزامن استخدام المعطيات في كتل المعطيات.

القيمة الافتراضية لا تتجاوز 255 حجوم الكتل.

لنفترض مثلاً أن قيمة MAXTRANS =3 وهنالك تحويل متزامن يحاول الوصول إلى كتلة تمّ الوصول إليها من قبل ثلاثة تحويلات فعالة. فالتعليمة في التحويل الرابع لا يمكنها الاستمرار حتى يقوم أحد التحويلات الثلاثة السابقة بإجراء عملية تثبيت Commit أو تراجع RollBack.

كمثال على الوسيطين MAXTRANS , INITRANS: لنفرض أنّ لدينا جدولاً كبيراً جداً وهنالك عدد قليل من المستخدمين الذين يقوموا بعمليات وصول متزامن إلى هذا الجدول، بالتالي فإن عدد التحويلات المتزامنة يكون قليل بالتالي يمكن وصف قيم صغيرة للوسيطين MAXTRANS , INITRANS.



مثال عن حساب وسائط التخزين Storage Parameter:

Create Tablespace test_storage
(...)
Storage (INITIAL 100k NEXT 100k
MINEXTENTS 2 MAXEXTENTS 5
PCTINCREASE 50);

ولنفرض أن قيمة الوسيط DB-BLOCK-SIZE = ٢k

يوضح الجدول التالي كيفية حجز المدى لهذا الفضاء الجدولي :

رقم الد .ى	د يم المدى الحالي	د يم المدى التالي
1	100K OR 50 BI ƆCKS	*100k*
2	100K OR 50 BI ƆCKS	*CEIL(100*1.5) : 150K*
3	150K OR 75 BI ƆCKS	*CEIL(150*1.5) : 228K*
4	228K OR 114 BI ƆCKS	*CEIL(228*1.5) : 342K*
5	342K OR 171 BI ƆCKS	*CEIL(342*1.5) : 516K*

إنشاء الفضاءات الجدولية

إنشاء فضاء جدولي باستخدام الأداة Server Manager

يمكن باستخدام تعليمة Create Tablespace إنشاء فضاء جدولي جديد وذلك على الشكل:

CREATE TABLESPACE tablespace
DATAFILE [filespec[,filespec]]
[DEFAULT STROAGE storage_clause]
[ONLINE|OFFLINE]
[PERMANENT|TEMPORARY]

حيث:

☆ **tablespace:** اسم الفضاء الجدولي الذي سيتم إنشاؤه.

☆ **DATAFILE:** لتحديد ملفات معطيات الفضاء الجدولي.

☆ **filespec:** تحديد توصيف ملفات المعطيات وتأخذ الشكل:

'filename' [SIZE n] [K or M] [REUSE]
[AUTOEXTEND OFF] | [AUTOEXTEND ON
[NEXT integer [K|M]]
[MAXSIZE UNLIMITED] |

[MAXSIZE integer [K|M]]
[NOLOGGING | LOGGING]

حيث:

- **filename**: اسم ملف المعطيات.
- **SIZE**: حجم الملف بالكيلوبايت K أو بالميغابايت M.
- **REUSE**: للسماح لمخدم أوراكل بإعادة استخدام الملفات الموجودة.
- **AUTOEXTEND OFF**: لعدم السماح بالتوسيع التلقائي لملفات المعطيات.
- **AUTOEXTEND ON**: للسماح بالتوسيع التلقائي لملفات المعطيات بتحديد قيمة المدى التالي NEXT عند امتلاء ملفات المعطيات.
- **MAXSIZE**: لتحديد الحجم الأعظم لملفات المعطيات إما بقيمة غير محددة UNLIMITED، أو بحجم محدد بالقيمة integer.
- **LOGGING**: لتسجيل معلومات الإرجاع Redo Log المتعلقة بالعمليات على الجداول والفهارس. وهو الخيار الافتراضي أما الخيار المعاكس فهو NOLOGGING.

☆ **DEFAULT STORAGE**: لتحديد وسطاء التخزين الافتراضية الخاصة بالفضاء الجدولي، وهي تأخذ الشكل:

([INITIAL integer [K|M]]
[NEXT integer [K|M]]
[MINEXTENTS integer]
[MAXEXTENTS integer]
[PCTINCREASE integer])

حيث:

- **INITIAL**: لتحديد حجم أول مدى سيتم حجزه في الفضاء الجدولي.
- **NEXT**: لتحديد حجم المدى التالي الذي سيتم حجزه.
- **MINEXTENTENTS**: لتحديد العدد الأصغر من المدى التي سيتم حجزها.
- **MANEXTENTENTS**: لتحديد العدد الأكبر من المدى التي سيتم حجزها.

- **PCTINCREASE**: لتحديد نسبة تزايد حجم المدى ابتداء من المدى الثاني.

☆ **ONLINE**: لجعل الفضاء الجدولي متاح مباشرة بعد إنشاء الفضاء الجدولي.

☆ **OFFLINE**: لجعل الفضاء الجدولي غير متاح بعد إنشاء الفضاء الجدولي.

☆ **PERMANENT**: لاستخدام الفضاء الجدولي للتعامل مع العناصر الدائمة، وهو الخيار الافتراضي.

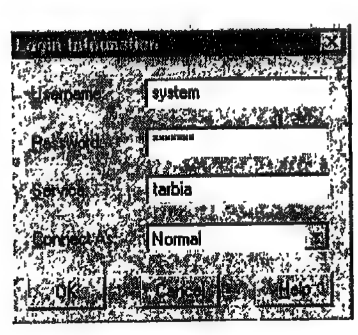
☆ **TEMPORARY**: لاستخدام الفضاء الجدولي للتعامل مع العناصر المؤقتة.



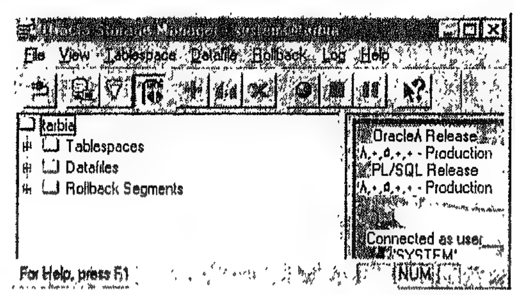
```
CREATE TABLESPACE ts_mohib
DATAFILE 'c:\mohib\tsmh1.dbf' SIZE 5M
DEFAULT STORAGE        (INITIAL 100K
                        NEXT 100K
                        MINEXTENTS 10
                        MAXEXTENTS 121
                        PCTINCREASE 25);
```

إنشاء فضاء جدولي باستخدام الأداة Storage Manager

يمكنك إنشاء فضاء جدولي باستخدام Storage Manager وذلك فقط مع هيئة أوراكل الفعالة. بعد تشغيل هذا البرنامج، تظهر نافذة تسجيل الدخول، أدخل اسم الحساب وكلمة المرور واسم الخدمة في حال كنت تقوم بتشغيل هذه الأداة من حاسب زبون (انظر الشكل ٢١-١).



الشكل ٢١-١

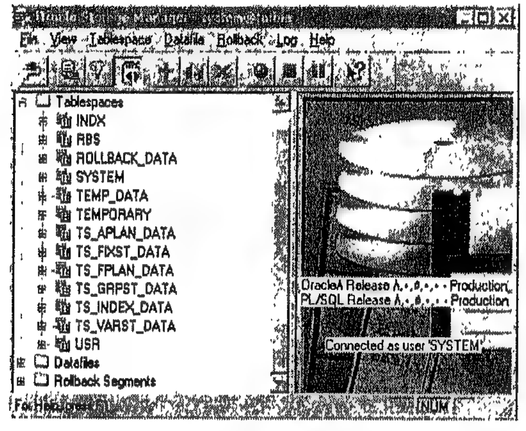


الشكل ٢١-٢

انقر زر OK ستظهر النافذة الرئيسية كما في الشكل ٢١-٢.

قم بتوسيع العقدة Tablespaces، تظهر شجرة فرعية تحتوي أسماء الفضاءات الجدولية الموجودة في القاعدة.

طبعا الفضاء الجدولي SYSTEM سيظهر ضمن هذه الشجرة، انظر الشكل ٢١-٣.

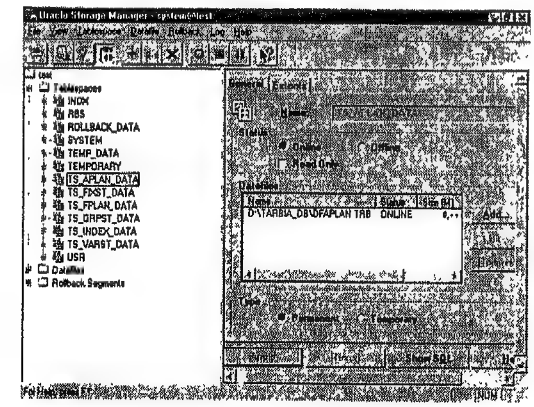


الشكل ٢١-٣

حدد أيا من هذه الفضاءات، تظهر النافذة اليمينية وهي تحتوي على معلومات عن هذا الفضاء:

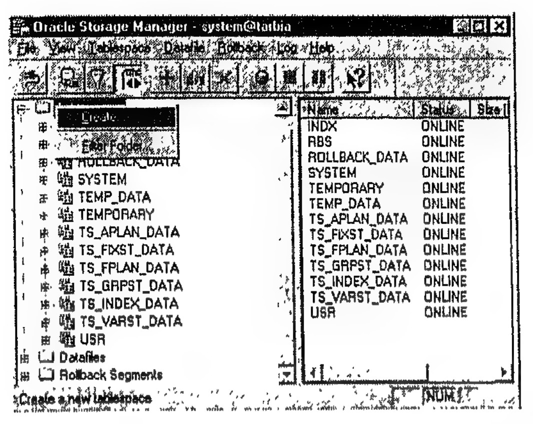
- **NAME**: اسم الفضاء الجدولي.
- **STATUS**: حالة هذا الفضاء فعالة ONLINE أو غير فعالة OFFLINE.
- **SIZE** : حجم الفضاء الجدولي.
- **USED** : المساحة المستخدمة من هذا الفضاء.

انظر الشكل٢١-٤.



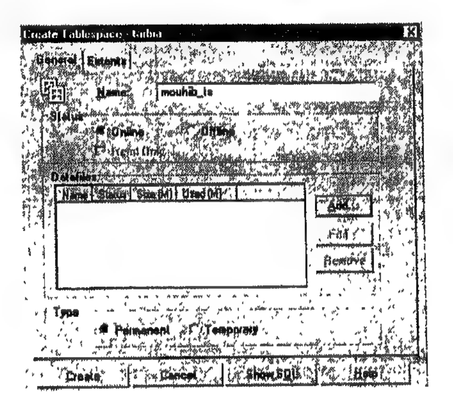
الشكل ٢١-٤

انقر الآن بزر الفأرة الأيمن على العقدة Tablespace تظهر لك قائمة متدلية تحتوي على الأوامر التي تستطيع إجراءها على الفضاء الجدولي. انظر الشكل ٢١-٥.



الشكل ٢١-٥

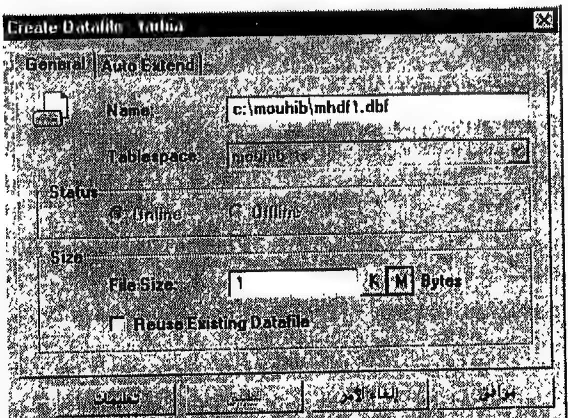
اختر Create، يظهر لك صندوق الحوار Create tablespace كما في الشكل ٢١-٦.



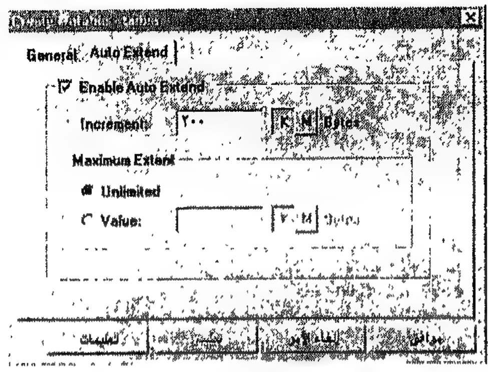
الشكل ٢١-٦

حدد اسم الفضاء الجدولي Name، وحالة الفضاء Status إما فعالة Online أو غير فعالة Offline، ثم قم بتحديد ملفات المعطيات Datafiles الخاصة بهذا الفضاء وذلك بالنقر على زر ADD، يظهر صندوق الحوار Create Datafile كما في الشكل ٢١-٧.

في صفحة التبويب General، حدد اسم ملف المعطيات Name وحجم هذا الملف File Size.



الشكل ٢١-٧

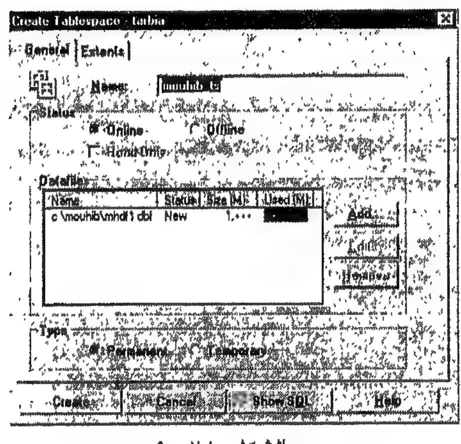


الشكل ٢١-٨

أمـــا ضـــمن صـــفحة التـــبويب Autoexetend، فتستطيع تفعيل خيار التوسيع التلقائي لملف المعطيات، حيث يمكـــنك تحديـــد مقـــدار الـــتزايد Increment، كذلـــك تحديـــد العـــدد الأعظم للمدى Maximum Extent، انظر الشكل ٢١-٨.

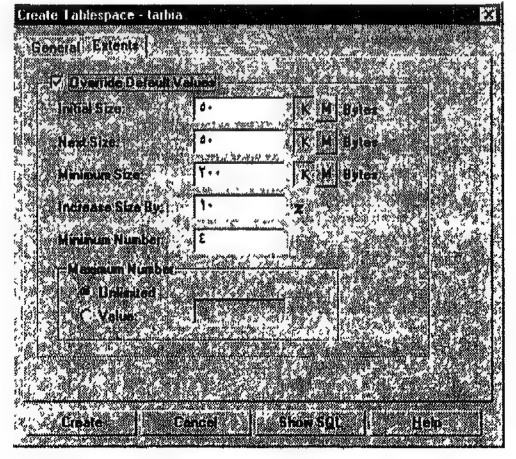
انقـــر زر OK فـــتظهر معلومات ملف المعطيـــات هـــذا في قائمة Datafiles (انظـــر الشكل ٢١-٩)، تستطيع إضافة مـــلفات معطيات جديدة بالنقر على زر ADD.

يمكنك أيضاً تحديد نمط الفضاء الجدولي Type إمـــا فضـــاء جدولـــي دائـــم Permanent أو فضـــاء جدولي مؤقت Temporary.



الشكل ٢١-٩

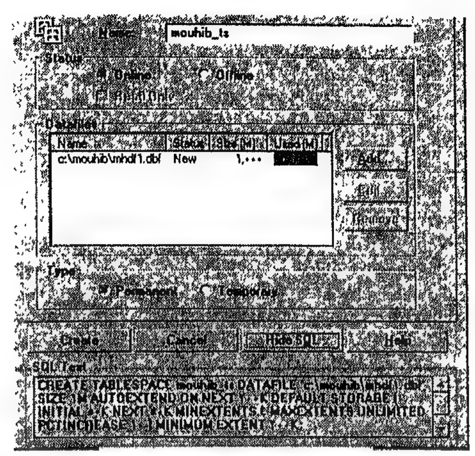
أمـــا في صفحة التبويب Extents، فتســـتطيع تحديد قيم وسطاء التخزين Storage Parameters والـــتي تمّ شـــرحها فـــي الفقرة السابقة، انظر الشكل ٢١-١٠.



الشكل ٢١-١٠

يمكنك أيضا رؤية تعليمة SQL الموافقة لإنشاء الفضاء الجدولي بالنقر على زر Show SQL.

انظر الشكل ٢١-١١.



الشكل ٢١-١١

أخيرا قم بعملية تسجيل إنشاء الفضاء الجدولي على قاعدة المعطيات بالنقر على زر Record في شريط الأدوات، انظر الشكل ٢١-١٢.

الشكل ٢١-١٢

تعديل الفضاءات الجدولية

يمكنك تعديل أي فضاء جدولي تم إنشاؤه من قبل وذلك باستخدام عدة طرق هي:

تعديل فضاء جدولي باستخدام الأداة Server Manager

يمكـن باسـتخدام تعليمة ALTER TABLESPACE تعديل فضاء جدولي وذلك على الشكل:

```
ALTER TABLESPACE tablespace
[ADD DATAFILE [filespec[,filespec]]]
[RENAME DATAFILE 'filename'[,'filename'] TO
                 'filename'[,'filename']
[DEFAULT STROAGE storage_clause]
[ONLINE]
[OFFLINE [NORMAL|TEMPORARY|IMMEDIATE]]
[BEGIN|END BACKUP]
[READ [ONLY|WRITE]]
[PERMANENT|TEMPORARY]
```

حيث:

☆ **tablespace**: اسم الفضاء الجدولي الذي سيتم تعديله.

☆ **ADD DATAFILE**: لإضافة ملفات معطيات جديدة بالتوصيف filespec إلى الفضاء الجدولي.

☆ **RENAME DATAFILE**: لتغيير أسماء ملفات معطيات الفضاء الجدولي.

☆ **DEFAULT STORAGE**: لـتحديد وسـطاء التخزين الافتراضية الخاصة بالفضاء الجدولي.

☆ **ONLINE**: لجعل الفضاء الجدولي فعالاً.

☆ **OFFLINE**: لجعل الفضاء الجدولي غير فعال.

☆ **NORMAL**: لإجـراء نقطـة تدقيق Checkpoint على جميع ملفات معطيات الفضاء الجدولي.

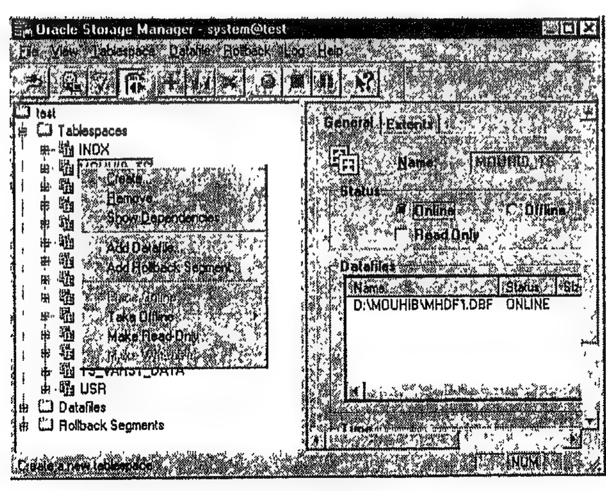
- ☆ TEMPORARY: لإجراء نقطة تدقيق فقط على ملفات معطيات الفضاء الجدولي الفعالة.
- ☆ IMMEDIATE: لإلغاء نقطة التدقيق وعدم إتاحة كافة ملفات المعطيات.
- ☆ BEGIN BACKUP: لتحضير الفضاء الجدولي للنسخ الاحتياطي الفعال.
- ☆ END BACKUP: لإعادة الفضاء الجدولي الذي تتم عليه عملية النسخ الاحتياطي إلى وضعه الطبيعي.
- ☆ READ ONLY: لمنع التغييرات على عناصر الفضاء الجدولي.
- ☆ READ WRITE: للسماح بإجراء عمليات إنشاء وتعديل وحذف عناصر الفضاء الجدولي.
- ☆ PERMANENT: لاستخدام الفضاء الجدولي للتعامل مع العناصر الدائمة، وهو الخيار الافتراضي.
- ☆ TEMPORARY: لاستخدام الفضاء الجدولي للتعامل مع العناصر المؤقتة.



```
ALTER TABLESPACE ts_mohib
ADD DATAFILE 'C:\mohib\tsmh2.dbf' SIZE 1M
READ ONLY;
```

تعديل فضاء جدولي باستخدام الأداة Storage Manager

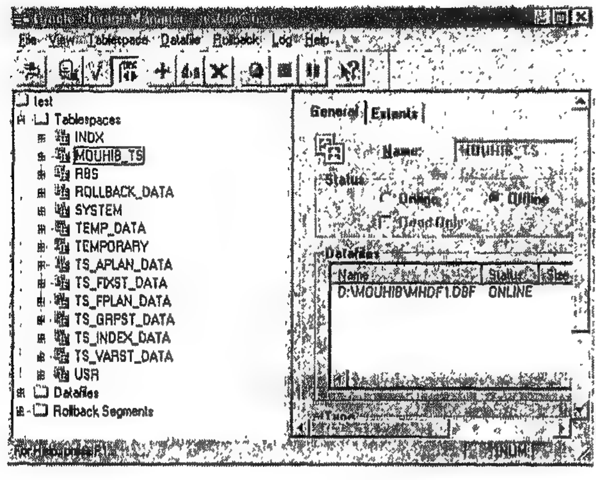
افتح نافذة Storage Manager، ثم قم بتوسيع العقدة Tablespaces تظهر شجرة فرعية تمثل الفضاءات الجدولية الموجودة في قاعدة المعطيات الحالية. انقر بزر الفأرة الأيمن على الفضاء الجدولي المطلوب تعديله (انظر الشكل ٢١-



الشكل ٢١-١٣

١٣) تظهر قائمة متدلية تستطيع من خلالها إجراء بعض التعديلات على الفضاء الجدولي كإضافة ملف معطيات جديد ..Add Datafile، أو إلغاء تفعيله Take Offline، أو جعله للقراءة فقط Make Read Only.

لاحظ أنه في الجزء الأيمن من النافذة تظهر المعلومات المتعلقة بالفضاء الجدولي المحدد. تستطيع من خلال هذا الجزء إجراء تعديلات على هذا الفضاء انظر الشكل ٢١-١٤.



الشكل ٢١-١٤



ما هي فائدة الفضاءات الجدولية المؤقتة Temporary Tablespaces؟

تستخدم الفضاءات الجدولية المؤقتة من أجل إجراء عمليات الفرز Sort التي لا يمكن إدراجها في الذاكرة. حيث يتم إنشاء واستخدام مقاطع مؤقتة Temporary segment وتقوم هذه المقاطع بدورها بحجز كتل مدى جديدة عند الحاجة. ويجب ملاحظة أنه لايمكن للفضاءات الجدولية المؤقتة احتواء عناصر دائمة، كذلك لايمكن إدارتها من قبل مدير القاعدة DBA.

حذف الفضاءات الجدولية

يمكنك حذف أي فضاء جدولي تم إنشاؤه من قبل وذلك باستخدام عدة طرق هي:

حذف فضاء جدولي باستخدام الأداة Server Manager

يمكن باستخدام تعليمة DROP TABLESPACE حذف فضاء جدولـــي وذلـــك علـــى الشكل:

DROP TABLESPACE tablespace
[INCLUDING CONTENTS [CASCADE CONSTRAINTS]]

حيث:

- ☆ **tablespace**: اسم الفضاء الجدولي المراد حذفه.
- ☆ **INCLUDING CONTENTS**: لحذف جميع محتويات الفضاء الجدولي.
- ☆ **CASCADE CONSTRAINTS**: لحذف جميع قيود التكامل المرجعي مـــن جداول خارج الفضاء الجدولي تدل على مفاتيح أولية في جداول الفضاء الجدولي الذي سيتم حذفه.



DROP TABLESPACE ts_mohib
INCLUDING CONSTRAINTS;

حذف فضاء جدولي باستخدام الأداة Storage Manager

افتح نافذة Storage Manager، ثم قم بتوسيع العقدة Tablespaces تظهر شجرة فرعيـة تمثل الفضاءات الجدولية الموجودة في قاعدة المعطيات الحالية. انقر بزر الفـــأرة الأيمـــن على الفضاء الجدولي المطلوب حذفه، ثم اطلب الأمر Remove من القائمة المتدلية (انظر الشكل ٢١-١٣)

بعض النصائح الخاصة بإدارة الفضاءات الجدولية

١. حاول إنشاء عدة فضاءات جدولية ضمن قاعدة المعطيات لأنها تعطيك مرونة أكثر في القيام بالعمليات على القاعدة، وبشكل خاص عند القيام بالمهام التالية :

- فصل معطيات المستخدم عن معطيات قاموس المعطيات.
- فصل معطيات تطبيق عن معطيات تطبيق آخر.
- تخزين ملفات معطيات فضاءات جدولية مختلفة على سواقات أقراص منفصلة مما يسمح بتقليص عمليات الإدخال والإخراج I/O.
- فصل معطيات مقاطع التراجع عن معطيات المستخدم مما يمنع ضياع المعطيات في حال حدوث عطل ما في قرص التخزين.
- تخصيص كل فضاء جدولي بنمط محدد من استخدامات قاعدة المعطيات كعمليات التعديل المتكررة أو عمليات القراءة أو عمليات التخزين المؤقتة.
- عند القيام بالنسخ الاحتياطي لأحد الفضاءات، فإنك تستطيع العمل مع بقية الفضاءات.

٢. حاول التقليل قدر الإمكان من عدد ملفات المعطيات ضمن الفضاء الجدولي وذلك من أجل تقليل عدد الملفات المطلوب فتحها في الوقت نفسه. طبعا يجب إنشاء ملف معطيات واحد على الأقل في الفضاء الجدولي SYSTEM.

٣. حدد حجوم ملفات المعطيات، ويجب أن يكون حجم ملف المعطيات الأول (في الفضاء الجدولي SYSTEM) 2M على الأقل لأنه سيحتوي على المعطيات الأولية للقاموس ومقطع التراجع.

٤. حاول تحديد مواقع ملفات المعطيات للحصول على التجاوب الأمثل، فكما تعلم فإن موقع الفضاء الجدولي يتحدد بالموقع الفيزيائي لملفات المعطيات التي تكونه. لذلك استخدم مصادر التخزين في حاسبك بالشكل الأمثل، فإذا كان لديك عدة سواقات أقراص فإنه من الأفضل تخزين جداول المعطيات في فضاء جدولي على أحد الأقراص، والفهارس على قرص آخر مما يساعد في الحصول على المعطيات من الجداول والفهارس في نفس الوقت.

٥. يجب عدم تخزين ملفات المعطيات وملفات الإرجاع Redo Log Files على نفس قرص التخزين.

٦. يجب تحديد وسطاء التخزين Storage Parameters عند إنشاء فضاء جدولي جديد. وحاول عدم أخذ القيم الافتراضية.
٧. يجب تحديد الامتيازات للمستخدمين كل حسب عمله.

الفضاءات الجدولية وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للفضاءات الجدولية هي:

☆ المشاهد المتعلقة بالفضاءات الجدولية: dba_tablespaces، user_tablespaces.

☆ المشاهد المتعلقة بالمدى: dba_extents، user_extents.

☆ المشاهد المتعلقة بالملفات: dba_data_files، v$datafiles.

☆ المشاهد المتعلقة بالمساحات الفارغة في الملفات: dba_free_space، user_free_space.



إدارة مقاطع التراجع

Administrating Rollback Segments

تحتوي أي قاعدة معطيات على مقطع أو أكثر من مقاطع التراجع، وهي عبارة عن جزء من القاعدة يحتوي على تسجيلات لحركات التحويلات التي يجب التراجع عنها في ظروف خاصة.

تستخدم مقاطع التراجع للقيام بالعمليات الأساسية التالية:

١. تناسق القراءة Read Consistency.

٢. التراجع عن التحويلات Rollback Transactions.

٣. استرداد قاعدة المعطيات Recover the database.

محتوى مقاطع التراجع

تتكون المعلومات الموجودة ضمن مقطع التراجع من عـــدة مداخــل تراجــع rollback entries. يحتوي كل مدخل من مداخل التراجع على معلومات عديدة، أهمــها معلومــات الكتلة (رمز الملف File ID ورمز الكتلة block ID الموافق للمعطيات التــي تغـيرت)، إضافةً إلى المعطيات التي كانت موجودة قبل أي عملية ضمن التحويل.

ترتبط جميع مداخل التراجع المتعلقة بنفس التحويل مع بعضها بحيـــث يمكــن الوصــول بسهولة إلى هذه المداخل عند الضرورة.

لا يمكن قراءة مقاطع التراجع من قبل مستخدمي قاعدة المعطيات أو مدير القاعدة. وإنمــا تكتب وتقرأ من قبل نظام أوراكل فقط ويمتلكها المستخدم SYS فقط بغض النظــر عــن المستخدم الذي أنشأها.

وباعتبار أن مداخل التراجع تقوم بتغيير كتل المعطيات، فإنها تسجل أيضــاً فــي ذواكــر الإرجاع وهذا مهم جداً بالنسبة للتحويلات الفعالة التي لم تحفظ وقــت حــدوث انــهيار أو توقف النظام حيث يتم التراجع عنها تلقائياً عند إعادة تشغيل الهيئــة أو إجــراء اســترداد الوسائط media recovery.

يقوم أوراكل بإنشاء جدول تحويلات transaction table لكل مقطع تراجع موجــود فــي القاعدة. يحتوي هذا الجدول على قائمة بالتحويلات التي تستخدم مقطع الــتراجع الموافــق ومداخل التراجع لكل تغيير يتم عمله من قبل هذه التحويلات.

تحتوي مقاطع التراجع على المعطيات قبل التغيير، وفي كل تحويل فإنّ أي تغيــير جديــد يرتبط بالتغيير السابق. فإذا احتاج التحويل إلى إجراء تراجع فإن التغييرات تطبق على كتل المعطيات بترتيب يؤدي إلى إعادة المعطيات إلى وضعها السابق.

كذلك عندما يحتاج أوراكل لتزويدنا بمجموعة قراءة متناسقة read-consistent كنتيجــة لاستعلام. فإنه يستطيع استخدام المعلومات في مقاطع التراجع لإنشاء مجموعــة معطيــات متماسكة.

يمكن بشكل خارجي ربط إجراء مع مقطع تراجع محدد من قبــل التطبيــق. فــي بدايــة التحويل، يمكن للمستخدم أو المطور تحديد مقطع تراجع خاص يمكن لأوراكل اســتخدامه عند تنفيذ التحويل. هذا يسمح للمستخدم أو للمطور باختبار مقطع تراجع صغير أو كبــير حسب التحويل.

كيف يتم إنشاء واستخدام المدى من قبل مقاطع التراجع؟

عندما يتم إنشاء مقطع تراجع، يمكن تحديد وسطاء التخزين للتحكم بكيفية حجز مدى لـهذا المقطع حيث يتم حجز كتلتي مدى على الأقل لكل مقطع تراجع.

يقوم التحويل بالكتابة وبشكل تسلسلي في مقطع تراجع وحيد وفي كتلة مدى واحـدة فقـط وذلك في وقت معين.

بالإضافة إلى ذلك يمكن للعديد من التحويلات الفعالة الكتابة بشكل متزامن في مقطع تراجع وحيد، وحتى في نفس المدى ضمن مقطع التراجع.

عندما يعمل التحويل خارج الفضاء المتاح في المدى الحالي ويحتاج لاستكمال عملية الكتابة يقوم أوراكل بإيجاد مدى جديد في مقطع التراجع، وهو يعتمد أحد الخيارين التاليين:

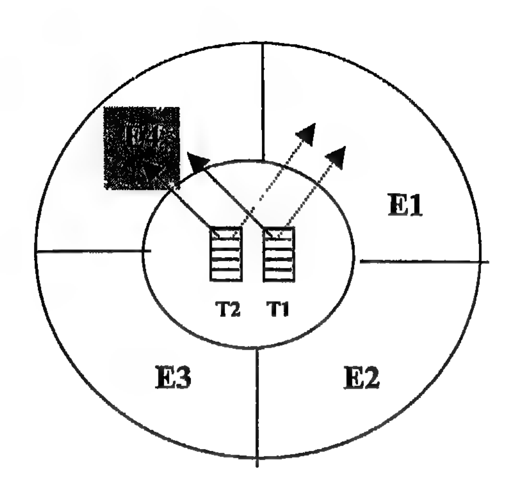
- إعادة استخدام مدى تم حجزه مسبقاً في مقطع التراجع.
- القيام بإنشاء مدىً جديد ضمن مقطع التراجع.

يقوم أول تحويل يحتاج إلى حجز مكان إضافي ضمن مقطع التراجع باختبار المدى التـالي ضمن المقطع، فإذا كان هذا المدى لا يحتوي على معلومات إرجاع فعالة، يقـوم أوراكـل بجعلها المدى الحالي، وتقوم جميع التحويلات بالكتابة في هذا المدى.

يوضح الشكل ٢٢-١ تحويلين T1, T2 يقومان بمتابعة الكتابة في المدى بشـكل دائـري متسلسل.





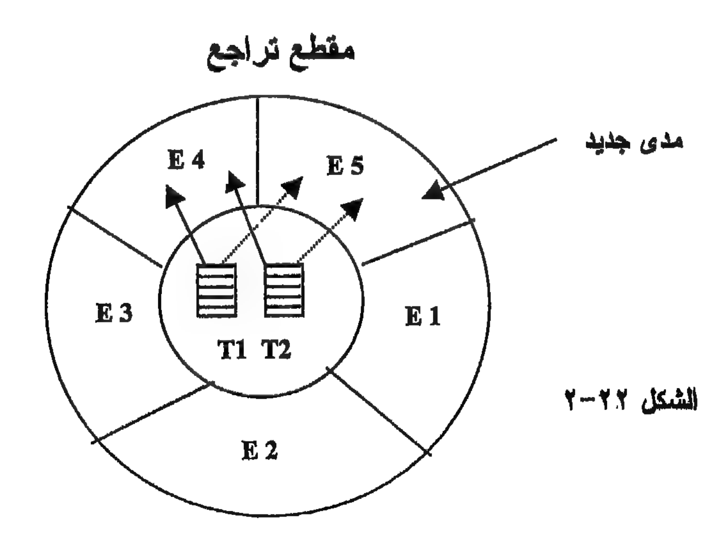


■ مدى فعال جديد بمساحة فارغة.

□ مدى فعال بمساحة ممتلئة.

الشكل ٢٢-١

من أجل متابعة كتابة معلومات التراجع للتحويل، يقوم أوراكل بمحاولـة إعـادة اسـتخدام المدى التالي في الحلقة. فإذا كانت هذه الكتلة تحتوي على معطيات فعالة، يقـوم أوراكـل بحجز مدى جديد حيث يمكنه حجز كتل مدى جديدة حتى يصل إلى قيمة وسيط التخزيـن MAXEXTENTS (انظر الشكل ٢٢-٢):



الشكل ٢٢-٢

كيف يتم إلغاء حجز المدى من مقاطع التراجع؟

عـــندما يتم إنشاء أو تعديل مقطع تراجع فإنه يمكن استخدام وسيط التخزين OPTIMAL (يستخدم فقط مع مقاطع التراجع) من أجل تحديد الحجم الأمثل لمقطع التراجع.

فـــإذا احتاج التحويل إلى متابعة كتابة معلومات التراجع من مدى إلى مدىً آخر في مقطع التراجع، يقوم أوراكل بمقارنة الحجم الحالي لمقطع التراجع مع الحجم الأمثل للمقاطع. فإذا كان مقطع التراجع أكبر من الحجم الأمثل وكانت كتل المدى اللاحقة للمدى الحالي الممتلئ غيـــر فعالــــة، يقوم أوراكل بإلغاء حجز هذه الكتل وبشكل متكرر حتى يصبح الحجم الكلي لمقطع التراجع مساوٍ أو قريب من الحجم الأمثل لكن ليس أقل منه.

وعـــندما يـــتم حـــذف مقطع تراجع، تتم إعادة جميع كتل المدى الخاصة بهذا المقطع إلى الفضاء الجدولي ويصبح بذلك متاحاً لمقاطع أخرى في الفضاء.

ماهو الارتباط بين هيئة أوراكل وأنماط مقاطع التراجع؟

عـــندما تقوم هيئة أوراكل بفتح قاعدة معطيات، يجب عليها جلب مقطع أو أكثر من مقاطع التراجع حتى تقوم بمعالجة معلومات التراجع الواردة من التحويلات.

ويمكنها التعامل مع نوعين من مقاطع التراجع:

١- **مقاطع التراجع الخاصة Private rollback segment**: ويتم جلبها بشكل خارجي مـــن قـــبل هيـــئة أوراكـــل عـــندما يقـــوم بفتح القاعدة، لذلك يجب تضمينها في الوسيط ROLLBACK_SEGMENT الموجـــود في ملف الوسطاء ليتم التعرّف عليها من قبل النظام.

٢- **مقـــاطع التراجع العامة Public rollback segment**: والتي تشكل حوض مقاطع التراجع الذي يمكن استخدامه مع أي هيئة من هيئات أوراكل.

ويتم إنشاء مقطع تراجع افتراضي بالاسم SYSTEM عند إنشاء قاعدة معطيات. وذلك في الفضـــاء الجدولي SYSTEM باستخدام قيم وسطاء التخزين الافتراضية المحددة في هذا الفضاء.

لا يمكن حذف هذا المقطع الافتراضي، وتقوم هيئة أوراكل بشكل دائم باستخدام هذه المقطع إضافة إلى أية مقاطع تراجع أخرى يحتاجها.

يمكن أن يكون مقطع التراجع في أحدى الحالات التالية:

- **OFFLINE**: لم يستخدم من قبل أي هيئة.
- **ONLINE**: استخدم من قبل هيئة أوراكل، ويمكن أن يحتوي على معطيات عن التحويلات الفعالة.
- **NEEDS RECOVERY**: تحتوي على معلومات عن التحويلات غير المثبتة uncommitted التي لا يمكن استردادها (لأن ملفات المعطيات لا يمكن الوصول إليها) أو تمّ إلغاؤها.
- **PARTLY AVAILABLE**: تحتوي على معطيات عن التحويلات الموزّعة التي لم يتمّ حلها.
- **INVALID**: أي تمّ حذفها.

إنشاء مقاطع التراجع

إنشاء مقاطع التراجع باستخدام الأداة Server Manager

يمكن باستخدام تعليمة CREATE ROLLBACK SEGMENT إنشاء فضاء جدولي جديد وذلك على الشكل:

```
CREATE      [PUBLIC|PRIVATE]
            ROLLBACK SEGMENT rollback_segment
[TABLESPACE tablespace]
[STORAGE (
        [INITIAL integer [K|M]]
        [NEXT integer [K|M]]
        [OPTIMAL integer [K|M]]
        [MINEXTENTS integer]
        [MAXEXTENTS integer])];
```

حيث:

- **rollback_segment**: اسم مقطع التراجع.

* **tablespace**: اسم الفضاء الجدولي، وفي حال لم يتم تحديد اسم الفضاء الجدولي يتم إنشاء مقطع التراجع في الفضاء الجدولي SYSTEM.

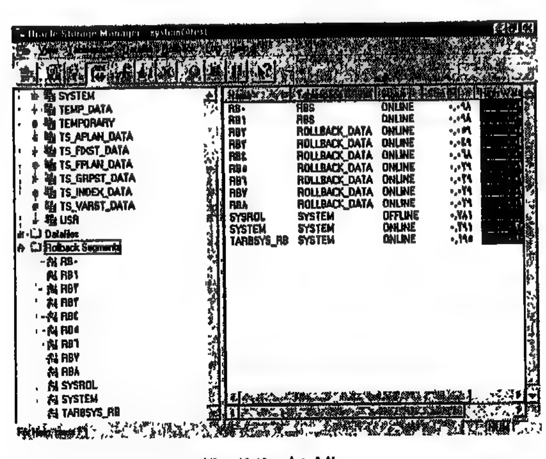
أما وسطاء التخزين الافتراضية فلقد تمّ شرحها في الفصل السابق، عدا الوسيط OPTIMAL فهو خاص بمقاطع التراجع ويمثل الحجم الأمثل لمقاطع التراجع والذي سيتم الرجوع إليه دوماً عندما لايصبح هناك حاجة لاستخدام كل المدى الموجودة فيها.



```
CREATE ROLLBACK SEGMENT rbs_mohib
TABLESPACE ts_mohib
STORAGE
        (INITIAL 10k
        NEXT 10k
        MINEXTENTS 20
        MAXEXTENTS 121
        OPTIMAL 200k);
```

إنشاء مقاطع التراجع باستخدام الأداة Storage Manager

يمكنك إنشاء مقطع تراجع باستخدام Storage Manager وذلك فقط مع هيئة أوراكل الفعالة. بعد تشغيل هذا البرنامج، ستظهر النافذة الرئيسية كما في الشكل ٢٢-٣.



الشكل ٢٢-٣

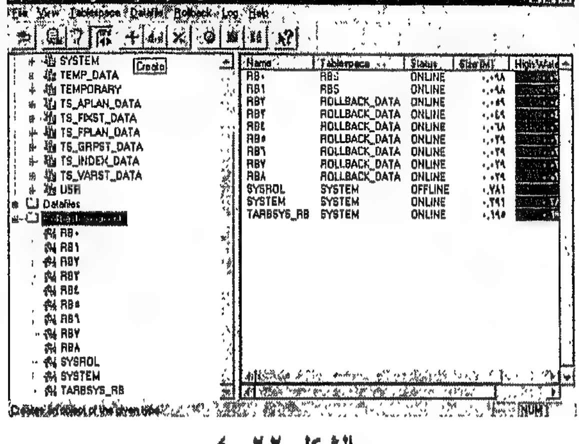
قم بتوسيع العقدة Rollback Segment، تظهر شجرة فرعية تحتوي أسماء مقاطع التراجع الموجودة في القاعدة، طبعاً مقطع التراجع SYSTEM سيظهر ضمن هذه الشجرة. حدّد أياً من هذه المقاطع، تظهر النافذة اليمينية وهي تحتوي على معلومات عن هذا المقطع:

- **Name**: اسم مقطع التراجع.

- **Tablespace**: اسم الفضاء الجدولي الذي يحتوي على مقطع التراجع.
- **Status**: حالة هذا المقطع فعّال ONLINE أو غير فعّال OFFLINE.
- **Size** : حجم مقطع التراجع.
- **High Water Mark** : المساحة المستخدمة من هذا الفضاء.

انقر الآن بزر الفأرة الأيمن على العقدة Rollback Segment تظهر لك قائمة متدلية تحتوي على الأوامر التي تستطيع إجراءها على مقاطع التراجع.

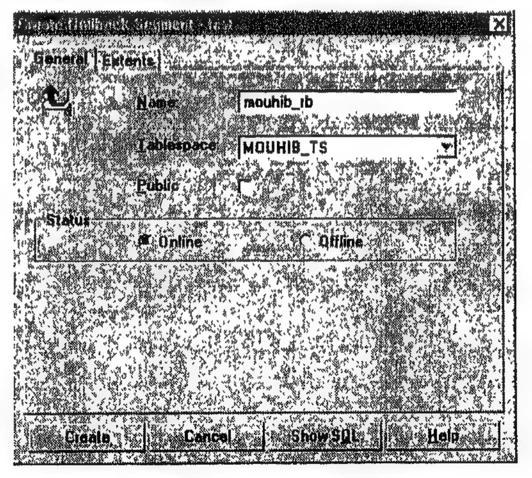
يمكن أيضاً القيام بإنشاء مقطع تراجع جديد بالنقر على زر Create في شريط الأدوات (انظر الشكل ٢٢-٤).



الشكل ٢٢-٤

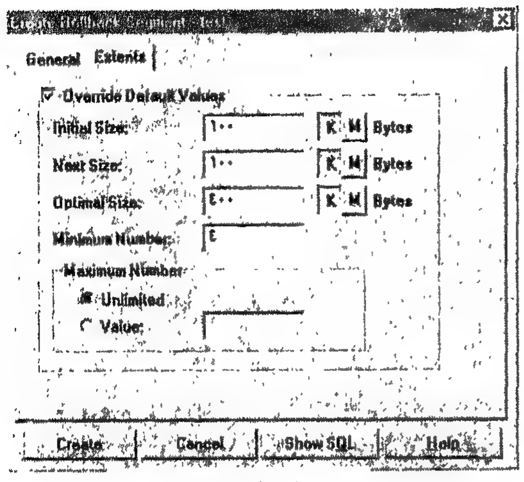
يظهر لك صندوق الحوار Create Rollback Segment كما في الشكل٢٢-٥.

حدّد اسم مقطع التراجع Name، واسم الفضاء الجدولي Tablespace وحالة المقطع Status إما فعّالة Online أو غير فعّالة Offline ،



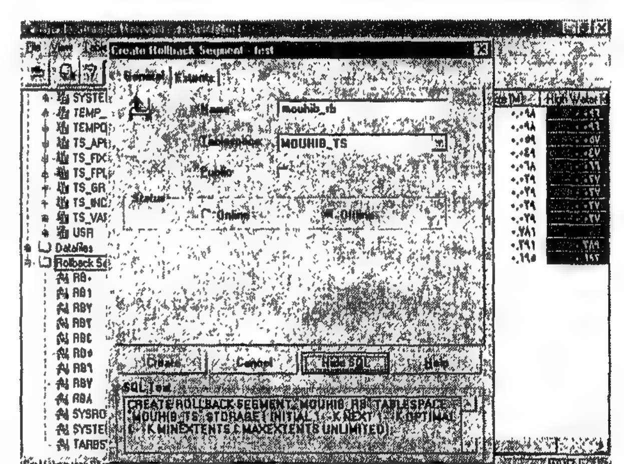
الشكل ٢٢-٥

انقر على زر التبويب Extents، تظهر مجموعة حقول تمكّنك من تحديد قيم وسطاء التخزين الخاصة بهذا المقطع (انظر الشكل ٢٢-٦).



الشكل ٢٢-٦

يمكنك رؤية تعليمة SQL الموافقة لإنشاء مقطع التراجع بالنقر على زر Show SQL.
انظر الشكل ٢٢-٧.



الشكل ٢٢-٧

تعديل مقاطع التراجع

يمكنك تعديل أي مقطع تراجع تم إنشاؤه من قبل وذلك باستخدام عدة طرق أهمها:

تعديل مقطع تراجع باستخدام الأداة Server Manager

يمكن باستخدام تعليمة ALTER ROLLBACK SEGMENT تعديل مقطع تراجع وذلك على الشكل:

```
ALTER ROLLBACK SEGMENT rollback_segment
[ONLINE]
[OFFLINE]
[STORAGE (
        [INITIAL integer [K|M]]
        [NEXT integer [k|M]]
        [OPTIMAL integer [K|M]]
        [MINEXTENTS integer])]
 [SHRINK [TO integer [K|M]]];
```

حيث يمكّننا الخيار SHRINK من إعادة تغيير حجم المقطع إلى الحجم الأمثل المحدد بالوسيط OPTIMAL أو إلى حجم محدد بالقيمة integer.

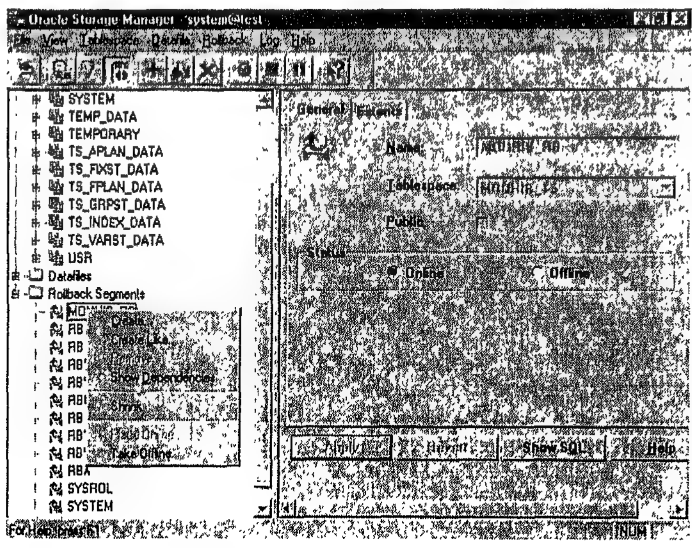
```
ALTER ROLLBACK SEGMENT rbs_mohib
ONLINE;
```



تعديل مقطع تراجع باستخدام الأداة Storage Manager

افتح نافذة Storage Manager، ثم قم بتوسيع العقدة Rollback Segment تظهر شجرة فرعية تمثل مقاطع التراجع الموجودة في قاعدة المعطيات الحالية. انقر بزر الفأرة الأيمن على المقطع المطلوب تعديله، تظهر قائمة متدلية تحتوي على بعض أوامر التعديل كأمر

Shrink وأمر Take Offline (انظر الشكل ٢٢-٨). تستطيع أيضاً إجـــراء التعديـــلات على مقطع التراجع بتحديد هذا المقطع وإجراء التعديلات المطلوبة في الجزء الأيمن مــــن النافذة.



الشكل ٢٢-٨

حذف مقاطع التراجع

يمكنك حذف أي مقطع تراجع تم إنشاؤه من قبل وذلك باستخدام عدة طرق أهمها:

حذف مقاطع التراجع باستخدام الأداة Server Manager

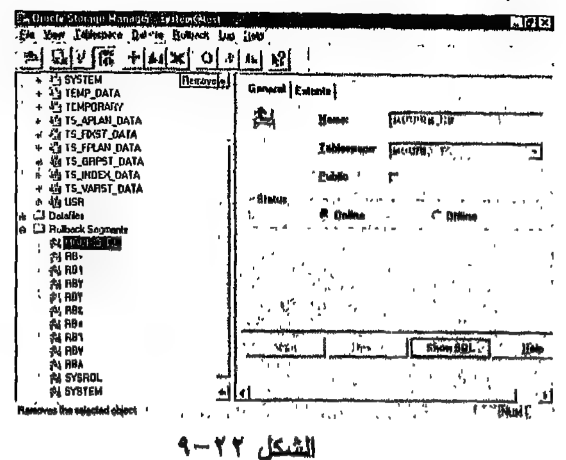
يمكن باستخدام تعليمة DROP ROLLBACK SEGMENT حذف مقطع تراجع وذلك على الشكل:

DROP ROLLBACK SEGMENT rollback_segment



DROP ROLLBACK SEGMENT rbs_mohib;

حذف مقاطع التراجع باستخدام الأداة Storage Manager


الشكل ٢٢-٩

افتح نافذة Storage Manager، ثم قم بتوسيع العقدة Rollback Segment تظهـــر شجرة فرعية تمـــثل مقاطع التراجع الموجودة في قاعدة المعطيات الحالية. حدّد مقطع التراجع المطلوب حذفه ثم انقــر عــلى زر Remove من شريط الأدوات (انظر الشكل ٢٢ -٩). طبعاً يجب أن يكون مقطع التراجع في حالة عدم تأهيل Offline، قبل حذفه.

مقاطع التراجع وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة لمقاطع التراجع هي:

* المشاهد المتعلقة بجميع المقاطع: dba_segments، user_segments.
* المشاهد المتعلقة بمقاطع التراجع: dba_rollback_segs.
* المشاهد المتعلقة بالتحويلات ومقاطع التراجع: v$rollname, v$transaction, v$session, v$rollstat.



لمعرفة بعض المعلومات عن مقاطع التراجع:

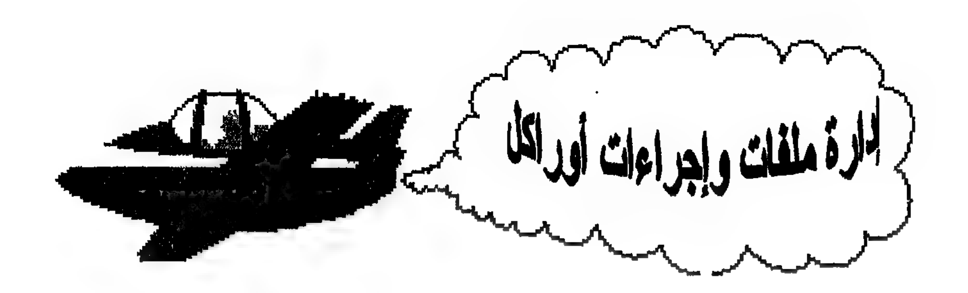
```
SELECT  segment_name,tablespace_name,status
FROM dba_rollback_segs;
```



لمعرفة التحويلات الفعالة في مقاطع التراجع:

```
SELECT  DISTINCT xacts "Active Transactions"
FROM v$rollname, v$rollstat vs
WHERE vs.usn=vs.usn and xacts!=0;
```


الباب السابع

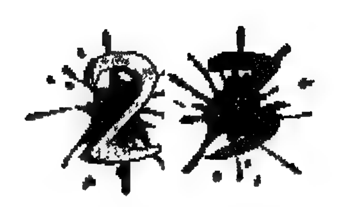


٢٣ . إدارة ملفات المعطيات.

٢٤ . إدارة ملفات التحكّم.

٢٥ . إدارة ملفات الإرجاع.

٢٦ . إدارة الإجراءات.





إدارة ملفات المعطيات
Administrating Data Files

كما ذكرنا سابقاً، تتكون قاعدة معطيات أوراكل من ملف أو أكثر من ملفات المعطيات، حيث تحتوي ملفات المعطيات على جميع معطيات القاعدة.

ويتم تخزين معطيات بنى قواعد المعطيات المنطقية كالجداول والفهارس بشكل فيزيائي ضمن ملفات المعطيات.

وعندما تتم قراءة معطيات من ملف معطيات يتم تخزينها في الذاكرة المخبئية لأوراكل. وبالتالي عندما يحتاج مستخدم إلى معلومات غير موجودة في الذاكرة المخبئية تتم قراءتها من ملف المعطيات المطلوب.

وليس ضرورياً أن تتم كتابة المعطيات الجديدة أو المعدّلة مباشرة في ملف المعطيات، وإنما يتم تجميع المعطيات في الذاكرة ومن ثم كتابتها في ملف المعطيات دفعة واحدة. طبعاً هذا

يقلل من عمليات الكتابة على القرص (الإجراء DBWR هو المسؤول عن عملية الكتابة هذه).

إضافة ملفات معطيات جديدة

يمكن إضافة ملفات معطيات جديدة إلى قاعدة معطيات باستخدام تعليمة ALTER TABLESPACE على الشكل:

ALTER TABLESPACE tablespace
[ADD DATAFILE [filespec[,filespec]]]

حيث:

☆ **tablespace**: اسم الفضاء الجدولي الذي سيتم تعديله.

☆ **ADD DATAFILE**: لإضافة ملفات معطيات جديدة بالتوصيف filespec إلى الفضاء الجدولي.

☆ **filespec**: توصيف ملف المعطيات الجديد ويأخذ الشكل:

```
'filename' [SIZE n] [K or M] [REUSE]
[AUTOEXTEND OFF] | [AUTOEXTEND ON
                    [NEXT  integer [K|M]]
                    [MAXSIZE UNLIMITED] |
                    [MAXSIZE integer [K|M]]
                    [NOLOGGING | LOGGING]
```

حيث:

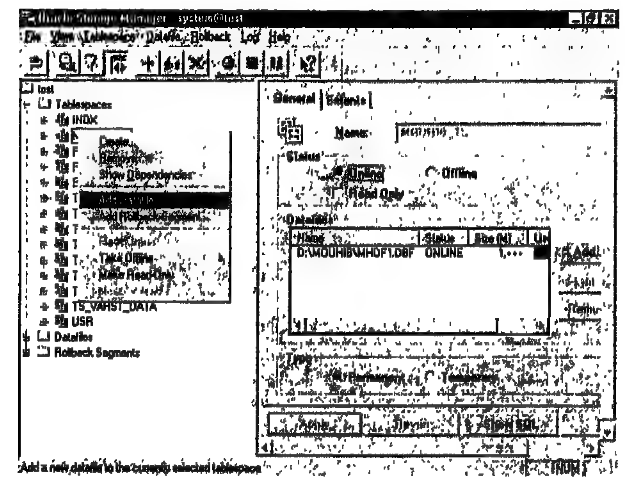
- **filename**: اسم ملف المعطيات.
- **SIZE**: حجم الملف بالكيلوبايت K أو بالميغابايت M.
- **REUSE**: للسماح لمخدّم أوراكل بإعادة استخدام الملفات الموجودة.
- **AUTOEXTEND OFF**: لعدم السماح بالتوسيع التلقائي لملفات المعطيات.
- **AUTOEXTEND ON**: للسماح بالتوسيع التلقائي لملفات المعطيات بتحديد قيمة التوسيع التالي NEXT عند امتلاء ملفات المعطيات.
- **MAXSIZE**: لتحديد الحجم الأعظم لملفات المعطيات إما بقيمة غير محددة UNLIMITED، أو بحجم محدد بالقيمة integer.

- **LOGGING**: لتسجيل معلومات الإرجاع Redo Log المتعلقة بالعمليات على الجداول والفهارس. وهو الخيار الافتراضي أما الخيار المعاكس فهو NOLOGGING.



ALTER TABLESPACE mohib
ADD DATAFILE 'mohdf.dbf' SIZE 10M;

تستطيع باستخدام الأداة Storage Manager إضافة ملفات معطيات جديدة. قم بتوسيع العقدة Tablespaces تظهر شجرة فرعية تمثل الفضاءات الجدولية الموجودة في قاعدة المعطيات الحالية. انقر بزر الفأرة الأيمن على الفضاء الجدولي المطلوب تعديله ثم اختر ...Add Datafile (انظر الشكل ٢٣-١).



الشكل ٢٣-١



تغيير اسم ملف معطيات

يمكن تغيير اسم ملف معطيات باستخدام التعليمة:

```
ALTER TABLESPACE tablespace
[RENAME DATAFILE 'filename'[,'filename'] TO
                 'filename'[,'filename']
```

حيث:

- ☆ **tablespace**: اسم الفضاء الجدولي الذي سيتم إنشاؤه.
- ☆ **RENAME DATAFILE**: لتغيير أسماء ملفات معطيات الفضاء الجدولي.



```
ALTER TABLESPACE mohib
RENAME DATAFILE 'mohdf.dbf' TO 'lamisdf.dbf';
```

تستطيع باستخدام الأداة Storage Manager تعديل اسم ملف معطيات. انقر بزر الفأرة الأيمن على ملف المعطيات المطلوب تغيير اسمه ثم اختر الأمر (انظر الشكل ٢٣-٢).

الشكل ٢٣-٢



تغيير حجم ملفات المعطيات تلقائياً

يمكن القيام بذلك باستخدام تعليمة ALTER TABLESPACE يوضح المثال التالي كيفية القيام بذلك:

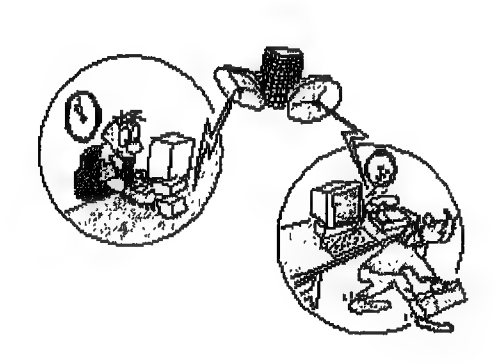


ALTER TABLESPACE mohib
ADD DATAFILE 'mohdf.dbf' SIZE 10M
AUTOEXTEND ON
NEXT 1M
MAXSIZE 100M;

يمكنك أيضاً تغيير حجم ملف معطيات باستخدام تعليمة ALTER DATABASE مثلاً:



ALTER DATABASE DATAFILE 'mohdf.dbf'
RESIZE 100M;



ملفات المعطيات وقاموس المعطيات

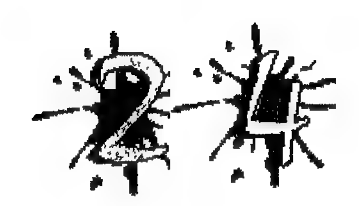
يحتوي قاموس المعطيات على العديد من المشاهد الموافقة لملفات المعطيات هي:

☆ المشاهد المتعلقة بالملفات: V$DATAFILES, DBA_DATA_FILES, V$DBFILE.



*SELECT * FROM dba_data_files;*







إدارة ملفّات التحكم
Administrating Control Files

تستخدم ملفات التحكم للاحتفاظ بالمعلومات الحرجة المتعلقة بالعمليات على نظام إدارة قواعد المعطيات العلاقاتية. وكما ذكرنا سابقاً تستخدم ملفات التحكم أثناء عملية إقلاع هيئة أوراكل لتحديد أسماء ومواقع ملفات المعطيات وملفات الإرجاع الخاصة بالقاعدة.

لذلك فإن أي ضياع في ملفات التحكم سيؤدي إلى تعطل عمل القاعدة، لذلك ينصح بشدة بالاحتفاظ بأكثر من نسخة من ملفات التحكم وعلى عدة أقراص تخزين بحيث يمكن استخدام أي من هذه النسخ في حال حدوث مشكلة في نسخة من نسخ ملفات التحكم.

تستطيع إضافة ملف تحكم جديد إلى القاعدة بعد أن يكون قد تمّ إنشاؤها باتباع الخطوات التالية:

١. قم بإطفاء هيئة أوراكل.

٢. قم بنسخ ملف التحكم إلى موقع آخر وعلى قرص تخزين مختلف.

٣. عدّل ملف الوسطاء وقم بإضافة ملف التحكم الجديد إلى الوسيط CONTROL_FILES.

٤. قم بإعادة تشغيل هيئة أوراكل.

الطريقة الأمثل لحفظ وحماية ملفات التحكم هي استخدام تعليمة ALTER DATABASE على الشكل:

ALTER DATABASE [database]
[BACKUP CONTROLFILE
[TO 'filename'[REUSE]] or [TO TRACE [RESETLOGS or
NORESETLOGS]]

حيث:

☆ **BACKUP CONTROLFILE**: يستخدم هذا الخيار لإنشاء نسخة احتياطية من ملف التحكم وذلك إما بتحديد اسم الملف الذي سيتم نسخه إليه 'To 'filename أو بكتابة تعليمات SQL إلى ملف الأثر الذي يمكن استخدامه لإعادة إنشاء ملفات التحكم TO TRACE. ويمكن فتح القاعدة باستخدام هذين الخيارين RESETLOG و NORESETLOG.

يمكن أيضاً إنشاء ملف تحكم جديد باستخدام تعليمة Create Controlfile على الشكل:

CREATE CONTROLFILE [REUSE][RESET]
DATABASE database
LOGFILE logfile_spec;
DATAFILE datafile_spec;









إدارة ملفات الإرجاع
Administrating Redo Log Files

ذكــرنا سابقاً، تستخدم ملفات الإرجاع لتخزين المعلومات المتعلقة بالإرجاع.

كما وفي كل مرة يتم تغيير المعطيات في قاعدة المعطيات، تتم كتابة سجل إرجاع يوصف هذه التغييرات.

باستخدام هذه المعلومات، يمكن استرجاع معلومات قاعدة المعطيات في حال حدوث عطل في النظام، كانقطاع التيار الكهربائي أو عطل أحد أجزاء النظام. وفي هذه الحالة يتم إيقاف ممثل أوراكل مما يؤدي إلى ضياع جميع المعطيات الموجودة في الذواكر المخبئية المؤقتة. ولا يتم حفظ سوى المعلومات المكتوبة على الأقراص.

وعــند إعادة إقلاع أوراكل، فإن المعلومات الموجودة في ملفات الإرجاع ستستخدم لتوليد الــتغييرات الــتي حدثــت عــلى قاعدة المعطيات، وبالتالي فإن جميع التحويلات المثبتة Committed transaction يــتم إرجاعهــا وهو ما يسمى بالإرجاع الأمامي Rolled

Forward أمـــا التحويلات التي أدت إلى تغيير المعطيات دون أن يتم تثبيتها فتتم إعادتها، وهو ما يسمى بالإرجاع الخلفي Rolled back.
لذلك فإن ملفات الإرجاع ضرورية جدا لاسترجاع المعلومات، وفي حال فقدان هذه الملفات فلن تستطيع أبدا استرجاع معلوماتك.

لكن كيف تعمل ملفات الإرجاع؟

أي تغيير يتم على قاعدة المعطيات يتم تسجيله في ملف الإرجاع، لذلك فإنه في حال حدوث عطل ما يمكن استرجاع جميع التغيرات التي تم إجراؤها منذ آخر عملية نسخ احتياطي.
عـــند إجراء عملية COMMIT يتم تسجيل معلومات الإرجاع في ذواكر الإرجاع المؤقتة redo log buffers ويقـــوم الإجــراء LGWR بكـــتابة المعلومات الموجودة في ذواكر الإرجـــاع المؤقـــتة وذلك في ملفات الإرجاع، ولا يتم إنهاء عملية COMMIT هذه قبل الانتهاء من كتابة معلومات الإرجاع في الملفات.
مجموعـــة ملفات الإرجاع log file Group عبارة عن مجموعة من ملفات الإرجاع التي يقـــوم أوراكـــل بنسخها تلقائيا mirrors، هذا يساعد على حماية ملفات الإرجاع في حال حدوث عطل ما في النظام. وفي حال استخدامك لخاصية مرآة القرص disk mirroring لحمايـــة مـــلفات الإرجـــاع، فليس من الضروري استخدام مجموعات ملفات الإرجاع لأن القـــرص قد تمت حمايته بالفعل ويكفي في هذه الحالة استخدام ملف إرجاع وحيد. ويفضل اســـتخدام ملفي إرجاع ، وعند امتلاء الملف الأول، فإن عملية تسجيل المعلومات تنتقل إلى المـــلف الثاني، وفي حال تأهيل الأرشفة Archiving وامتلاء الملف الثاني وحدوث log switch، يمكن أن يتم نسخ هذه الملفات في ملف الإرجاع المؤرشف archive log file.

نقـــاط الـــتدقيق وتحويـــل التسجيل Log Switch and Check Points

عندما يمتلئ ملف الإرجاع أو تمتلئ مجموعة الإرجاع، فإن التسجيل يقلب switches إلى ملف الإرجاع التالي في السلسلة وهو ما يسمى بتحويل التسجيل log switch ويؤدي إلى حدوث مجموعة أحداث معينة بشكل تلقائي:

نقاط التدقيق Check pointing

إن تحويل تسجيل log switch يؤدي دوماً إلى حدوث نقطة تدقيق. نقطة التدقيق هذه تقوم بـــتفريغ جميع الذواكر المخبئية المؤقتة الموجودة ضمن أوراكل. هذا يؤدي إلى تقليل زمن الاسترداد عند الحاجة إلى ذلك.

الأرشفة Archiving

فـــي حـــال تفعيـــل عملية الأرشفة فإن تحويل التسجيل يقوم بنسخ محتويات ملف الإرجاع الفعـــال إلى مـــلف إرجاع مؤرشف archive log file. ويمكن استخدام هذا الملف أثناء عملية الاسترجاع عند الحاجة.

رقم الإرجاع التسلسلي Log Sequence Number

فـــي كـــل مرة تتم فيها إعادة استخدام ملف إرجاع، فإنه يعطى رقم إرجاع تسلسلي، وهو يعـــطى أيضا لملف الإرجاع المؤرشف الموافق ويستطيع RDBMS من خلال هذا الرقم معرفة ملف الإرجاع وملف الإرجاع المؤرشف الذي تم استخدامه.

إدارة مجموعات وملفات الإرجاع باستخدام Enterprise Manager

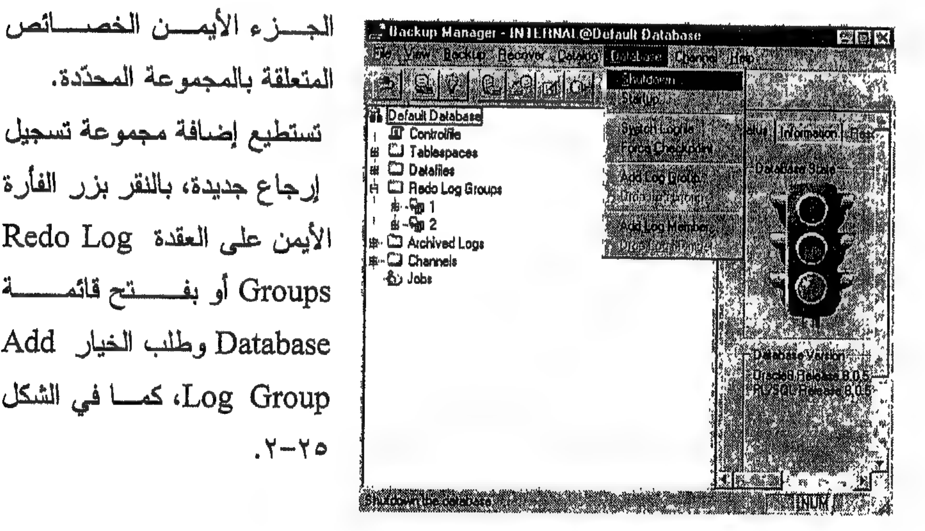
تستطيع باستخدام الأداة Enterprise Manager إدارة مجموعات وملفات الإرجاع. قم بتشغيل برنامج Backup Manager تظهر النافذة الرئيسية للبرنامج.

قم بتوسع العقدة Redo Log Groups، تظهر مجموعات تسجيل الإرجاع كما في الشكل ٢٥-١.

الشكل ٢٥-١

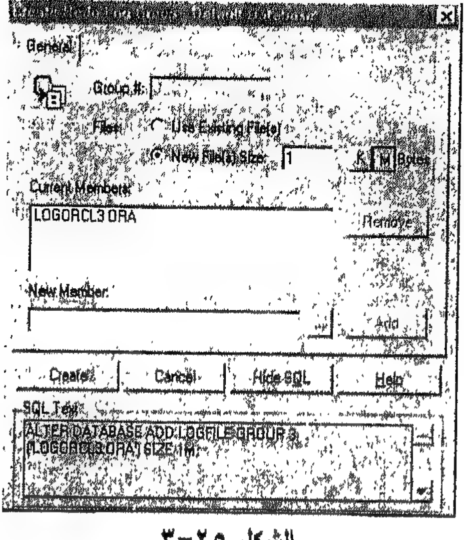
لاحظ في الجزء الأيمن الخصائص المتعلقة بالمجموعة المحدّدة.

تستطيع إضافة مجموعة تسجيل إرجاع جديدة، بالنقر بزر الفأرة الأيمن على العقدة Redo Log Groups أو بفتح قائمة Database وطلب الخيار Add Log Group، كما في الشكل ٢٥-٢.



الشكل ٢٥-٢

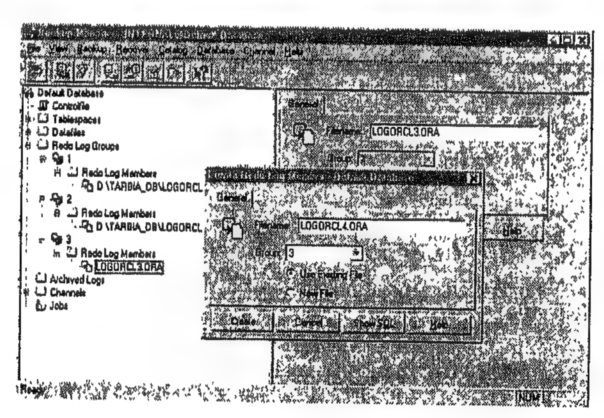
تظهر نافذة حوار كما في الشكل ٢٥-٣



الشكل ٢٥-٣

يمكنك من خلال هذه النافذة تحديد ملفات هذه المجموعة إما باستخدام ملفات موجودة مسبقاً Use Existing File (s) أو بإضافة ملفات جديدة يجب تحديد حجومها New File (s) Size . الآن لإضافة عضو جديد، انقر زر Add، الموجود بجانب الحقل New Member ، فيظهر صندوق الحوار Create Redo Log File، حدد اسم الملف name ورقم مجموعة الإرجاع التي سيتم وضع هذا الملف فيها، انقر زر OK فتتم إضافة العضو الجديد إلى قائمة الأعضاء الحاليين Current Members كما في الشكل ٢٥-٣.

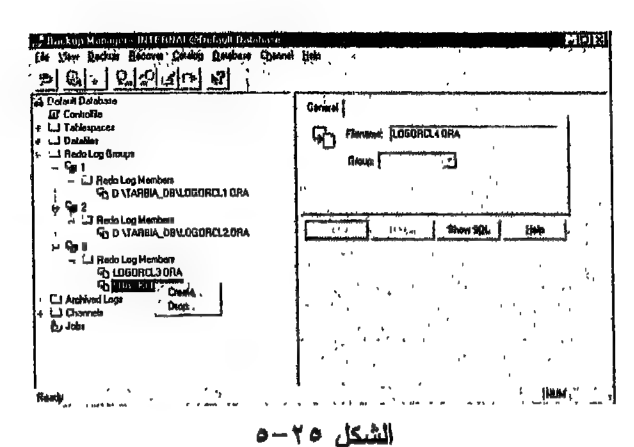
بعد النقر على زر Create، تظهر مجموعة تسجيل الإرجاع الجديدة ضمن قائمة المجموعات. يمكنك الآن إضافة عضو جديد إلى المجموعة السابقة بطلب الخيار Add Log Member من قائمة Database، يظهر صندوق حوار يشبه الشكل ٢٥-٤.



الشكل ٢٥-٤

حدّد اسم الملف المتعلّق بهذا العضو، إما من خلال ملف موجود مسبقاً Use Existing File أو بإضافة ملف جديد .

تستطيع أخيرا حذف مجموعة الإرجاع أو ملف الإرجاع المطلوب بالنقر بزر الفأرة الأيمن على المجموعة أو الملف المطلوب واختيار أمر Drop، انظر الشكل ٢٥-٥



الشكل ٢٥-٥

إدارة مجموعات وملفات الإرجاع باستخدام Server Manager

يمكن باستخدام تعليمة ALTER DATABASE إضافة أو تعديل مجموعات وملفات الإرجاع.

مثلاً لإضافة ملف إرجاع جديد:



```
ALTER DATABASE ORCL
ADD LOG FILE ('log 3a', 'log 3b') SIZE SM;
```

كما يمكن إضافة ملف إرجاع لمجموعة موجودة مسبقا، مثلاً:



```
ALTER DATABASE ORCL
ADD LOGFILE 'log 3c' To GROUP 1;
```

تستطيع أيضاً التعديل على مجموعات وملفات الإرجاع، يمكنك مثلاً تغيير اسم ملف إرجاع على الشكل:



```
ALTER DATABASE ORCLE
RENAME LOGFILE 'log 1' To 'log 2';
```

تستطيع أيضاً حذف ملف إرجاع على الشكل:



```
ALTER DATABASE ORCL
DROP LOGFILE 'log 1';
```

أما لحذف كامل مجموعة الإرجاع:

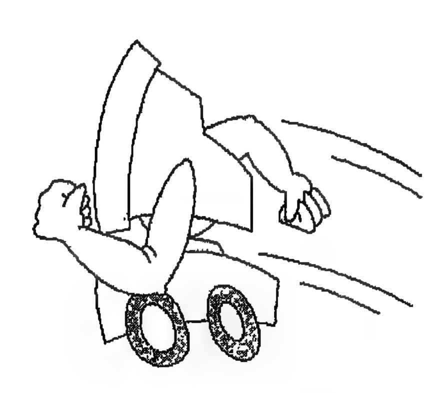


```
ALTER DATABASE ORCL
DROP LOGFILE GROUP 1;
```

تستطيع أيضاً حذف عضو من مجموعة إرجاع على الشكل:



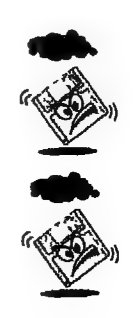
```
ALTER DATABASE ORCL
DROP LOGFILE GROUP MEMBER 'log 3a';
```



ملفات الإرجاع وقاموس المعطيات

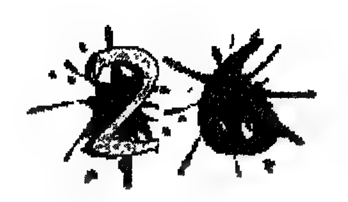
يحتوي قاموس المعطيات على العديد من المشاهد الموافقة لملفات الإرجاع هي:

- ☆ معلومات ذواكر تسجيل الإرجاع المؤقتة: V$SYSSTAT.
- ☆ معـلومات ملفات الإرجاع: V$LOGFILE, V$LOG, V$LOGHIST, V$LOG_HISTORY,V$RECOVERY_LOG.
- ☆ معلومات ملفات الإرجاع المؤرشفة: V$ARCHIVE.



```
SELECT name,value
FROM v$sysstat
WHERE name='redo log space requests';

SELECT *
FROM v$logfile;
```



إدارة الإجراءات
Administrating Processes

يمكنك في أغلب أنظمة الاستثمار تشغيل هيئة أوراكل إما في حالة إجراء وحيد single-process أو ضمن حالة إجراءات متعددة multiple-processes.

في الحالة الأولى فإن نظام أوراكل لا يقبل سوى مستخدماً واحداً في وقت معين. أما في الحالة الثانية فيمكنه قبول عدة مستخدمين في الوقت نفسه.

ويتم تحديد ذلك من خلال الوسيط SINGLE-PROCESS والذي يأخذ أحدى القيمتين TRUE أو FALSE.

كذلك يمكن تشغيل أوراكل بحالة مخدّم متعدّد التشعّبات Multi-threaded Servers وذلك بتحديد الوسطاء:

- *SHARED-POOL-SIZE*
- *MTS-LISTENER-ADDRESS*
- *MTS-SERVICE*

- ❖ *MTS-DISPATCHERS*
- ❖ *MTS-MAX-DISPATCHERS*
- ❖ *MTS-SERVERS*
- ❖ *MTS-MAX-SERVERS*

سنقوم فيما يلي بشرح كل وسيط منها:

١-**SHARED-POOL-SIZE**: لإضـــافة مســـاحات جديدة في الحوض المشترك Shared pool وذلــك مــن أجــل تخــزين معلومات عن الاتصالات بين إجراءات المستخدم، والموزّعين والمخدّمات، قم بإضافة1k لهذا الوسيط لكل مستخدم.

٢- **MTS-LISTENER-ADDRESS**: يجــب تحديد قيمة هذا الوسيط لكل بوابة ستقوم قاعدة المعطيات بالاتصال بها. تحديد هذا الوسيط يأخذ الشكل:

MTS-LISTENER-ADDRESS="(addr)"

حيــث addr هــو العــنوان الــذي سيقوم المستمع بالاستماع من خلاله لطلبات الاتصال لبروتوكول محدد.

يمكن أن يحتوي هذا الملف على عناوين عديدة مثل:

MTS-LISTENER-ADDRESS="(ADDRESS=(PROTOCOL=tcp) (PORT=5000) (HOST=ZEUS)"
MTS-LISTENER-ADDRESS="(ADDRESS=(PROTOCOL=decent) (OBJECT=OUTA)\ (NODE=ZEUS)"

٣-**MTS-SERVICE**: لتحديد اسم الخدمة المرتبطة مع الموزّعين.

فــإذا كان اسم خدمة الموزّع service dispatcher هو TEST-DB فإن تحديد هذا الوسيط يكون على الشكل:

MTS-SERVICE="test-db"

٤- **MTS-DISPATCHERS**: لــتحديد عــدد إجــراءات الموزّع dispatcher processes التي سيتم تشغيلها أثناء إقلاع هيئة القاعدة.

ويمكن توقع هذا العدد بالعلاقة:

عدد إجراءات المرسل= العدد الأعظم للدورات المتزامنة/ عدد الاتصالات لكل مرسل.

يمكن مثلاً تحديد هذا الوسيط على الشكل:

MTS-DISPATCHERS="TCPIP,3"

٥-**MTS-MAX-DISPATCHERS**: لــتحديد العدد الأعظم لإجراءات الموزّع التي يمكن تشغيلها خلال عمل هيئة قاعدة.

٦-MTS-SERVERS: لتحديد عدد إجراءات المخدّم التي يمكن تشغيلها أثناء إقلاع هيئة القاعدة. ويعتمد على العدد المتوقع للمستخدمين الذين سيتصلون بالقاعدة.

٧- MTS-MAX-SERVERS: لتحديد العدد الأعظم لإجراءات المخدّم التي يمكن تشغيلها أثناء إقلاع الهيئة.

يمكن تغيير قيمة أي وسيط من الوسطاء السابقة باستخدام التعليمة ALTER SYSTEM مثلاً:



ALTER SYSTEM
SET MTS-SERVERS=2

أو:



ALTER SYSTEM
SET MTS-DISPATCHERS="TCPIP, 4"

يمكن مراقبة إجراءات قاعدة معطيات أوراكل بإحدى طريقتين:

١-PROCESS: مراقب الإجراءات هذا يعطينا معلومات تلخيصية عن كل الإجراءات ضمن أوراكل كإجراءات المخدّم- الزبون، وإجراءات المستخدم وإجراءات المخدّم وإجراءات الخلفية،....إلخ.

٢-SESSION: مراقب الدورات هذا يعطينا رقم الدورة وحالة كل دورة.

- يقوم أوراكل باستخدام تقنية معينة لتسمية إجراءات هيئة، حيث أن أسماء إجراءات الخلفية تكون مسبوقة باسم الهيئة لتفريق إجراءات هيئة معينة عن إجراءات هيئة أخرى.

فإذا كانت لدينا هيئة بالاسم TEST، فإن إجراءات الخلفية الخاصة بها تكون:

- *ORA-TEST-DBWR*
- *ORA-TEST-LGWR*
- *ORA-TEST-SMON*
- *ORA-TEST-PMON*
- *ORA-TEST-RECO*
- *ORA-TEST-LCK0*
- *ORA-TEST-ARCH*
- *ORA-TEST-D000*

الإجراءات وقاموس المعطيات

- ☆ المشاهد المتعلقة بالإجراءات: V$DISPATCHER, V$QUEUE, V$SHARED_SERVER, V$THREAD.
- ☆ المشاهد المتعلقة بذاكرة أوراكل: VSVA, VDB_OBJECT_CACHE, V$OPEN_CURSOR.
- ☆ المشاهد المتعلقة بالتحويلات: V$TRANSACTION.
- ☆ المشاهد المتعلقة بالإحصائيات عن أعمال القاعدة: V$FILESTAT, V$LIBRARYCACHE, V$REQDIST, V$ROLLSTAT, V$ROWCACHE, V$SESSION_IO, V$SESSSTAT, V$STATNAME, V$WAITSTAT.

١. إظهار استخدام الذواكر وفق دورات المستخدم:

```
SELECT user_name, value || 'bytes' "Current session memory"
FROM v$session sess, v$sesstat stat, v$statname name
WHERE sess.sid = stat.sid
      AND stat.statistic# = name.statistic#
      AND name.name = 'session memory';
```

٢. إظهار مجموعة من الإحصائيات المفيدة:

```
SELECT name, value
FROM v$sysstat
WHERE name IN ('db block gets', 'consistent gets', 'physical
reads');
```

٣. معرفة رقم دورة والرقم التسلسلي لها بالاستفسار ضمن الجدول V$SESSION:



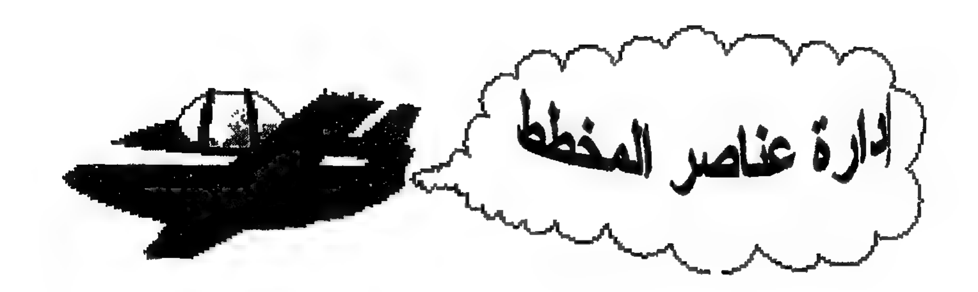
```
SELECT sid, serial#
      FROM v$session
      WHERE username="MOUHIB";
```

٤. تحديد حجم PGA لدورة:



```
SELECT ss.sid, ss.value, sn.name
FROM v$sysstat ss, v$statname sn, v$session se
WHERE ss.statistic#=sn.statistic#
      AND sn.name in ('session pga memory')
      AND se.sid=ss.sid
      AND type!='BACKGROUND';
```


الباب الثامن



٢٧ . إدارة الجداول.

٢٨ . إدارة المشاهد.

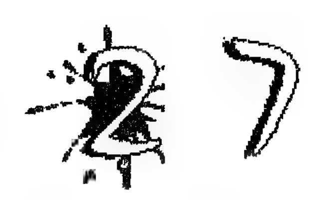
٢٩ . إدارة المرادفات.

٣٠ . إدارة الفهارس.

٣١ . إدارة السلاسل.

٣٢ . إدارة التجمعات.

٣٣ . إدارة الوحدات البرمجية.





إدارة الجداول
Administrating Tables

كما ذكــرنا فــإن الجــدول هو عبارة عن بنية تستخدم من قبل أوراكل لتخزين المعطيات.

ويــتكون كل جدول من مجموعة من الأسطر والأعمدة. حيث تمثل الأعمدة الحقول التي توصف الجدول، أما الأسطر فهي سجلات هذا الجدول.

ضــمن أوراكل، يتم إنشاء الجدول في مقطع جدول. ويتكون مقطع الجدول بدوره من مدىً أو أكثر، وعندما يكبر الجدول ويملأ المدى الحالية، يتم إنشاء مدىً جديد لهذا الجدول. ويتوسع المدى بالطريقة المحددة في عبارة STORAGE المستخدمة لإنشاء الجدول. وفي حــال عــدم تضــمين هذه العبارة أثناء إنشاء الجدول، يتم استخدام عبارة STORAGE الافتراضــية المعــرّفة في الفضاء الجدولي، وفي حال عدم تعريفها في الفضاء الجدولي، تؤخذ من القيم الافتراضية للنظام.

يتم تخزين معطيات الجدول في كتل المعطيات، ويرتبط عدد الأسطر التي يمكن تخزينها في كتلة معطيات واحدة بحسب حجم هذه الأسطر ووسطاء التخزين storage parameters ووسطاء إدارة المساحة الفارغة في كتل المعطيات Space Management Parameters.

وسطاء إدارة المساحة Space Management Parameters

يتشابه شكل كتل المعطيات بغض النظر فيما لو كانت هذه الكتل تحتوي على معطيات جدول أو معطيات فهرس أو معطيات تجمّع.

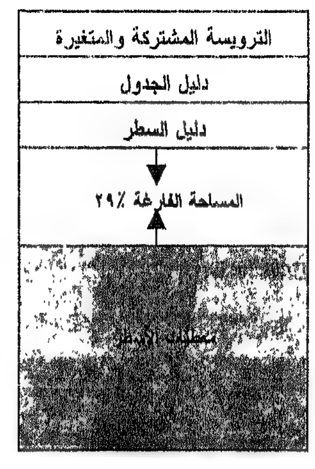
ويمكن أن تحتوي المساحة الفارغة في كتل المعطيات على مداخل للتحويلات التي يمكنها الوصول إلى سطر أو أكثر في الكتلة.

ويمكن التحكّم باستخدام المساحة الفارغة من أجل عمليات إدراج الأسطر أو تعديلها أو حذفها، وذلك بتحديد القيم الموافقة لوسطاء استخدام المساحة وهي:

- **PCTFREE**: لتحديد النسبة المئوية من مساحة كتل المعطيات المخصصة للتعديلات المستقبلية على أسطر الجدول. القيمة الافتراضية هي ١٠%.
- **PCTUSED**: لتحديد النسبة المئوية الصغرى للمساحة المستخدمة من كتل المعطيات. وتصبح الكتلة مرشّحة لإدراج أسطر جديدة عندما تصبح المساحة المستخدمة أقل من PCTUSED. القيمة الافتراضية هي ٤٠%.
- **INITRANS**: لتحديد عدد مداخل التحويلات المحجوزة مبدئياً ضمن ترويسة الكتلة.
- **MAXTRANS**: لتحديد العدد الأعظم من التحويلات التي يمكنها الوصول بشكل متزامن إلى كتلة المعطيات.

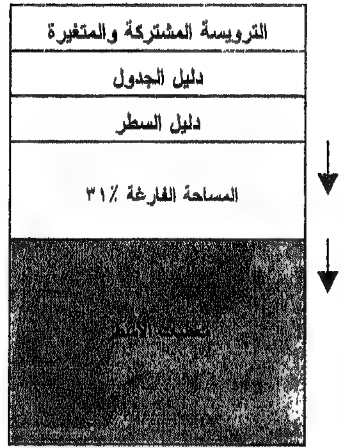
يمكن حساب الوسيط PCTFREE باستخدام الصيغة:

PCTFREE=(العدد الأعظم لبايتات كل سطر)-(عدد البايتات المدرجة في كل سطر)*100 / (العدد الأعظم لبايتات كل سطر)



الشكل ٢٧-١

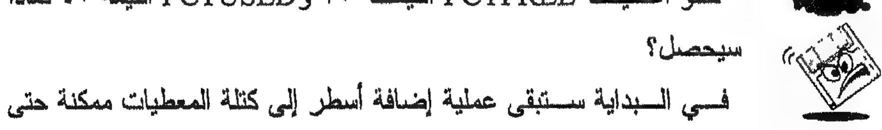
يوضح الشكل ٢٧-١ كتلة معطيات بقيمة وسيط PCTFREE=٣٠.



الشكل ٢٧-٢

أما الشكل ٢٧-٢ فيوضح كتلة معطيات بقيمة وسيط PCTUSED=٧٠.

عندما تصل النسبة المئوية للمساحة الفارغة في كتلة المعطيات القيمة PCTFREE، لايمكن إدراج أسطراً جديدة حتى تصبح النسبة المئوية للمساحة المستخدمة أقل من PCTUSED. لذلك إذا كانت قيمة الوسيط PCTUSED صغيرة، فإن كتل المعطيات لن تكون حرّة أغلب الأحيان.



لو أعطينا PCTFREE القيمة ٢٠ وPCTUSED القيمة ٤٠ فماذا سيحصل؟

في البداية ستبقى عملية إضافة أسطر إلى كتلة المعطيات ممكنة حتى وصول قيمة مساحة الفراغ المتبقي إلى ٢٠٪، عندها تتوقف الكتلة عن إضافة أسطر جديدة وتبقى في هذه الحالة حتى لو تمّ حذف الأسطر وكبر الفراغ إلى أن تنزل كمية المعلومات عن ٤٠٪. عندها فقط تفتح كتلة المعطيات أبوابها على مصراعيها

لتستقبل الأسطر الجديدة مفسحة لهم المجال ليملؤوا الفراغ وتعود مساحة الفراغ لتتقلص حتى الوصول إلى٪٢٠ وهكذا دواليك تتراوح كتلة المعطيات بين ٪٢٠ و ٪٤٠.

☆ PCTFREE المنخفض له التأثيرات التالية:

١- يقلص من المساحة المستهلكة.

٢- يزيد من زمن المعالجة.

٣- يقلص المساحة المتبقية للتعديل.

٤- يمكن للتعديلات الكثيرة أن تزيد وبشكل دراماتيكي من زمن المعالجة.

ولهذا تستخدم كتل معطيات بقيم صغيرة لهذا المعامل للعناصر الثابتة التي لا تتطلب تعديلات كثيرة.

☆ PCTFREE العالي له التأثيرات التالية:

١- يقلص من وقت المعالجة.

٢- معد للتعامل مع التعديلات.

٣- يستخدم في المعطيات الغير ثابتة.

☆ PCTUSED المنخفض له التأثيرات التالية:

١- يقلص المساحة المتاحة للمعلومات.

٢- يقلص الزمن اللازم للتعديل والحذف.

٣- يزيد من حجم المساحات الضائعة في قاعدة المعطيات.

☆ PCTUSED العالي له التأثيرات التالية:

١- يزيد من المساحات المستخدمة.

٢- يقلص حجم المساحات الضائعة في قاعدة المعطيات.

٣- يزيد من زمن المعالجة لعمليات الإدراج والحذف.

نصائح للأخذ بها...

- ☆ يجب أن لا يزيد مجموع PCTFREE وPCTUSED عن ١٠٠.
- ☆ كلما كبر المجموع كان هناك استخدام أفضل للمساحات على حساب زمن المعالجة.
- ☆ بالنسبة للجداول غير التجمعيّة : يفضل دائماً ترك فراغ يتراوح بين ٢٥٪ و ١٠٪ حسب إمكانية تغير المعلومات.
- ☆ بالنسبة للجداول التجمعيّة: يفضل ترك مساحات أكبر تصل إلى ٤٠٪.
- ☆ بالنسبة للفهارس: بما أن الفهارس لا تعدل إلا قليلاً جداً لذلك يمكن أن تكون ٥٪ أو أقل.



لدينا جدول تعديلاته كثيرة جداً وحجم السجلات يتراوح بصورة مهولة: يفضل أن يكون PCTTFREE حوالي ٢٠٪ وPCTUSED ٤٠٪.



لدينا جدول الأسطر فيه تحذف وتضاف بشكل كبير والتعديلات إن وجدت فحجم السجلات متقارب: PCTFREE ٥٪ و PCTUSED ٩٠٪.

لنفترض أن حجم كتلة المعطيات مساوٍ ٢٠٤٨ بايت، نطرح منها ١٠٠ بايت حجم الترويسة، يتبقى ١٩٤٨ بايت متاحة للمعطيات. فإذا افترضنا أن كل سطر يأخذ كحجم متوسط ١٩٥ بايت أو ١٠٪ من حجم المعطيات المتاح. لذلك فإن القيمة الأمثل لمجموع الوسيطين PCTFREE و PCTUSED يفضل أن تكون مساوية ٩٠٪ من أجل الاستخدام الأفضل لمساحة قاعدة المعطيات.

تحديد عدد التحويلات الفعّالة على كتل المعطيات

يمكن تحديد عدد التحويلات الفعّالة على كتلـــة معطيـــات وحيـــدة باســـتخدام الوســـيطين INITRANS و MAXTRANS.

وكما ذكرنا سابقاً فإنّ الوسيط INITRANS هو عدد مداخل التحويلات المبدئية المحجوزة في كل ترويسة كتلة للتحويلات المتزامنة. القيمة الافتراضية لهذا الوسيط هي ١ وتـــتراوح بين ١ و ٢٥٥. وتأخذ مداخل التحويلات حوالي ٢٣ بايت وفقاً لنظام التشغيل.

أما الوسيط MAXTRANS فهو العدد الأعظم للتحويلات التي يمكنها الوصـــول بشـــكل متزامن إلى كتلة المعطيات. القيمة الافتراضية لهذا الوسيط تختلف حسب نظام التشغيل لكن عادةً تأخذ القيمة ٢٥٥.

وكل تحويل يحتاج تقريباً ٢٣ بايت من المساحة الفارغة لكتلة المعطيات، فإذا تمّ اســـتخدام كامل المساحة الفارغة، سيضطر التحويل للانتظار حتى يستطيع الوصول إلى الكتلة.

يوضح الشكل ٢٧-٣ كيفية توصيف ترويسة كتل المعطيات.

ترويسة الكتلة

مؤشر سطر	مؤشر سطر	مؤشر سطر	مؤشر سطر	مؤشر سطر
مؤشر سطر	مؤشر سطر	مؤشر سطر	مؤشر سطر	مؤشر سطر
مؤشر سطر	مؤشر سطر			
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

الشكل ٢٧-٣

أما كتلة المعطيات فتأخذ الشكل ٢٧-٤:

ترويسة الكتلة			
معطيات للعمود٢	طول العمود٢	معطيات العمود١	طول العمود١
معطيات العمود٣		طول العمود٣	معطيات للعمود٢

الشكل ٢٧-٤

وفي حال لم يتسع سطر ما في كتلة معطيات يتم إنشاء سلسلة Chain بين عدة كتل معطيات بحيث تحتوي كل كتلة على مؤشر يدلّ على الكتلة التالية.

أما إذا تمّ تعديل سطر بحيث لم تعد المساحة المتبقية في الكتلة تتسع لهذا التعديل، فإنه فسيتم ترحيل Migration السطر إلى الكتلة المسلسلة بينما تبقى ترويسة السطر في الكتلة الأصلية.

طبعاً فإنّ أداء عمليات الإدخال والإخراج يقلّ عند قراءة أسطر مسلسلة أو مرحّلة، كونه يتم مسح أكثر من كتلة معطيات واحدة.

يمكن تكبير قيمة PCTFREE لتقليل تكرار ترحيل الأسطر، أو القيام بتصدير العناصر ثم حذفها ومن ثمّ إعادة استيرادها.

إدارة الجداول باستخدام Server Manager

يمكنك إجراء مختلف العمليات على الجداول باستخدام الأداة Server Manager.

إنشاء جدول جديد *Create Table*

يمكنك إنشاء جدول جديد باستخدام تعليمة CREATE TABLE التي تأخذ الشكل:

```
CREATE TABLE [schema, ] table
(column datatype [ DEFAULT expr] [ column_ constraint]
[,column datatype [ DEFAULT expr] [ column_ constraint]],
[table_constraint]);
[PCTFREE number]
[PCTUSED number]
[INITRANS number]
[MAXTRANS number]
[TABLESPACE tablespace]
[STORAGE
(
        [INITIAL number K or M]
        [NEXT number K or M]
        [MINEXTENTS number]
        [MAXEXTENTS number or MAXEXTENTS UNLIMITED]
        [PCTINCREASE number]
        [FREELISTS number]
        [FREELIST GROUP number]
        [OPTIMAL [number K or M ] or [NULL]]
)]
[CLUSTER cluster (column [, ... ])]
[PARALLEL parallel_clause]
[ENABLE enable_ clause]
[DISABLE disable_ clause]
[AS subquery]
[CACHE or NOCACHE]
```

حيث:

- **schema**: اسم المخطط الذي سيتم فيه إنشاء الجدول.
- **Table**: اسم الجدول.
- **Column**: اسم العمود.
- **Datatype**: نمط العمود.
- **PCTFREE**: كميـــة المســـاحة المحجوزة في كل كتلة (كنسبة مئوية من كامل المساحة عدا مساحة ترويسة الكتلة) وذلك كي تتوسع الأسطر طولياً.
- **PCTUSED**: تمثل الحد الأدنى من المساحة المستخدمة من الكتلة قبل أن تصبح متاحة لإدراج أسطراً جديدة.
- **INITRANS**: تمـثل عـدد مداخل التحويلات التي سيتم حجزها مسبقاً في كل كتلة. القيمة الافتراضية هي ١.
- **MAXTRANS**: تحديـد عـدد مداخل التحويلات التي يمكن حجزها لكل كتلة. القيمة الافتراضية هي ٢٥٥.
- **TABLESPACE**: لـتحديد اسـم الفضاء الجدولي الذي سيتم إنشاء الجدول ضمنه.
- **STORAGE**: لـتحديد عـبارة التخزين التي ستحدد كيفية حجز المدى الخاصة بالجدول.
- **RECOVERABLE**: لـتحديد تعـليمة إنشاء الجدول التي ستوضع في ملف الإرجاع وهي القيمة الافتراضية.
- **UNRECOVERABLE**: لعدم وضع تعليمة إنشاء الجدول في ملف الإرجاع.
- **CLUSTER**: لتحديد اسم التجميع الذي يؤلف الجدول جزءاً منه.
- **PARALLEL**: لتحديد خصائص التوازي الخاصة بالجدول.
- **ENABLE**: لتأهيل قيد التكامل.
- **DISABLE**: لإلغاء تأهيل قيد التكامل.
- **AS subquerey**: لإدراج الأسـطر الـتي تـم إرجاعهـا بالاستعلام الجزئي subquerey إلى الجدول بعد إنشائه.
- **CACHE**: يحدد هذا الوسيط أن الكتل المستحصلة من الجدول قد تم وضعها في أقرب نهاية للقائمة LRU موجودة في الذاكرة المخبئية عند مسح كامل الجدول.

- NOCACHE: يحدد هذا الوسيط أن الكتل المستخدمة من الجدول قد تم وضعها في أبعد نهاية للقائمة LRU موجودة في الذاكرة المخبئية عند مسح كامل الجدول.



```
CREATE TABLE test
(id number,
name VARCHAR2 (30 ) PRIMARY KEY,
age number )
PCTFREE 20  PCTUSED 75
STORAGE ( PCTINCREASE 0
      MAXEXTENTS 991)
TABLESPACE users;
```

تعديل بنية جدول Alter Table

أما لتعديل بنية جدول ما، فيمكنك استخدام تعليمة ALTER TABLE على الشكل:

```
ALTER TABLE [schema, ] table
[ADD ( column datatype [ DEFAULT expr ] [column_ constraint]
[column datatype [ DEFAULT expr ] [column_ constraint]]
[table_constraint])]
[MODIFY (column datatype [DEFAULT expr] [column_
constraint])]
[PCTFREE number]
[PCTUSED number]
[INITRANSE number]
[MAXTRANSE number]
[STORAGE storage_clause]
[DROP drop_clause]
[ALLOCATE EXTENT (     [ SIZE number K or M]
[DATEFILE datafile]
[INSTANCE number])
[DELLOCATE UNUSED [KEEP number K or M]]
[ENABLE enable_clause [TABLE LOCK]]
[DISABLE disable_clause [TABLELOCK]]
[PARALLEL parallel_clause [CACHE or NOCACHE]]
```

حيث:

حيث:

- **ALLOCATE EXTENT**: لإضافة مدىً جديد.
- **SIZE**: حجم المدى.
- **DATAFILE**: لتحديد اسم ملف المعطيات في الفضاء الجدولي.
- **INSTANCE**: رقم هيئة أوراكل.
- **DEALLOCATE UNUSED**: لإلغاء حجز المساحات غير المستخدمة.
- **KEEP**: لـتحديد المساحة التي سيتم إبقاؤها بعد إلغاء حجز المساحات غير المستخدمة.
- **ENABLE enable_clause**: لتأهيل قيد تكامل أو تأهيل جميع الإجراءات المرتبطة بجدول.
- **ENABLE TABLE LOCK**: لـتأهيل القفـل على تعليمات DDL و DML في بيئة مخدّم متوازي.
- **DISABLE disable_clause**: لإلغـاء تـأهيل قيد تكامل أو إلغاء تأهيل جميع الإجراءات المرتبطة بجدول.
- **DISABLE TABLE LOCK**: لإلغاء تأهيل القفل على تعليمات DDL و DML في بيئة مخدّم متوازي.



ALTER TABLE test
STORAGE (MAXEXTENTS / 2 /
PCTINCREASE 10)

إلغاء حجز المساحات التي لم تستخدم

يمكن إلغاء حجز المساحات غير المستخدمة، وذلك كي تصبح هذه المساحات متاحة لمقاطع أخرى باستخدام الخيار DEALLOCATE UNUSED مع تعليمة ALTER TABLE على الشكل:

ALTER TABLE [schema.]table
[DEALLOCATE UNUSED [KEEP integer[K|M]]];

حيث يتم إبقاء بعض المساحة الفارغة بتحديد قيمتها بعد الخيار KEEP.



ALTER TABLE employee
DEALLOCATE UNUSED;

حذف جدول Drop Table

يمكن حذف جدول باستخدام تعليمة Drop Table على الشكل :

DROP TABLE [schema.] table [CASCADE CONSTRAINTS]

حيث يفيد الخيار CASCADE CONSTRAINTS بحذف جميع قيود التكامل المرجعي والخاصة بالقيدين UNIQUE, PRIMARY KEY من الجدول.

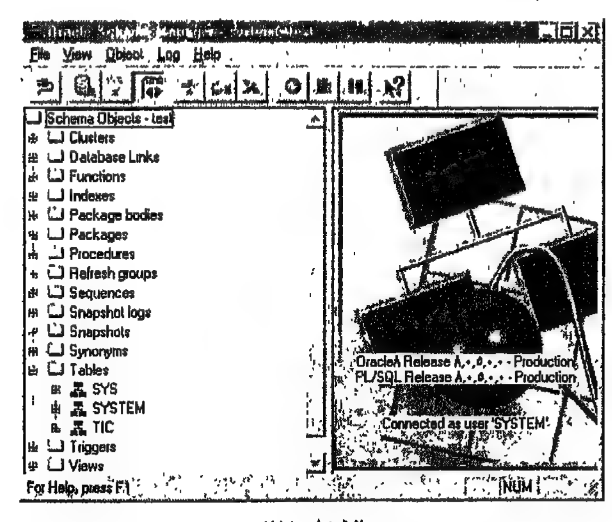


SQL > DROP TABLE test;

إدارة الجداول باستخدام الأداة Schema Manager

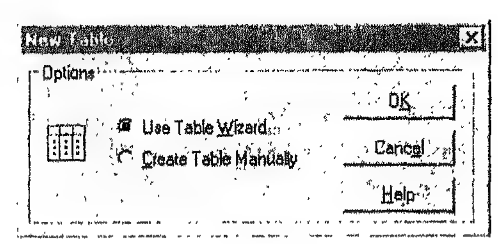
عندما تقوم بتشغيل برنامج Schema Manager تظهر قائمة عناصر المخطط المشابهة للشكل ٢٧-٥.

قم بتوسيع العقدة Table، تظهر لك قائمة مخططات المستخدمين user schemas، حدد مخطط العناصر الذي ترغب برؤية جداوله، ترى في الجزء الأيمن من النافذة، قائمة



الشكل ٢٧-٥

بأسماء الجداول الموجودة في هذا المخطط واسم الفضاء الجدولي الذي يتواجد فيه كل جدول.

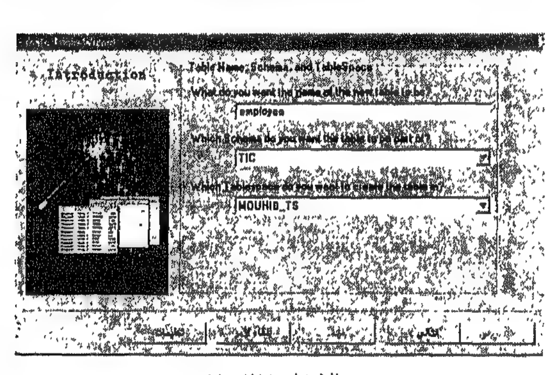


الشكل ٢٧-٦

من أجل إنشاء جدول جديد، انقر بزر الفأرة الأيمن على العقدة Table ثم اختر Create، يظهر صندوق الحوار New table، كما في الشكل ٢٧-٦.

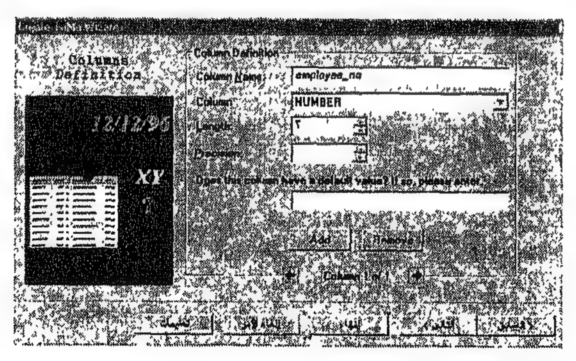
تستطيع من خلال هذا الصندوق تحديد طريقة إنشاء الجدول، باستخدام معالج الجداول Table Wizard أو بشكل يدوي Manually.

سنقوم حاليا باستخدام معالج الجداول، من أجل ذلك حدد الزر: Use Table Wizard ثم انقر زر OK، تظهر مجموعة من الصفحات التي تساعدك في إنشاء الجدول:



الشكل ٢٧-٧

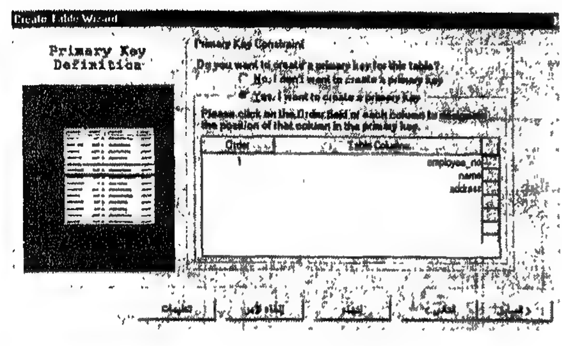
١- الصفحة الأولى (انظر الشكل ٢٧-٧)، تطلب منك اسم الجدول الذي سيتم إنشاؤه وضمن أي مخطط وأي فضاء جدولي.



الشكل ٢٧-٨

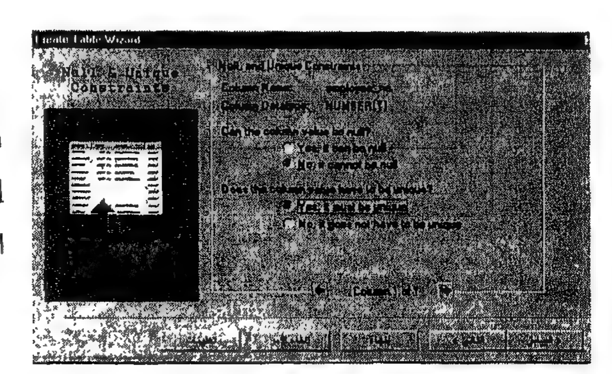
٢- أما الصفحة الثانية فتمكنك من تعريف أعمدة الجدول والقيمة الافتراضية لهذه الأعمدة، انظر الشكل ٢٧-٨.

٣- تمكنك الصفحة الثالثة من تعريف مفتاح أولي Primary Key للجدول، وفي حال كون المفتاح الأولي مؤلفاً من أكثر من عمود، تستطيع تحديد ترتيب Order هذه الأعمدة ضمن المفتاح، انظر الشكل ٢٧-٩.



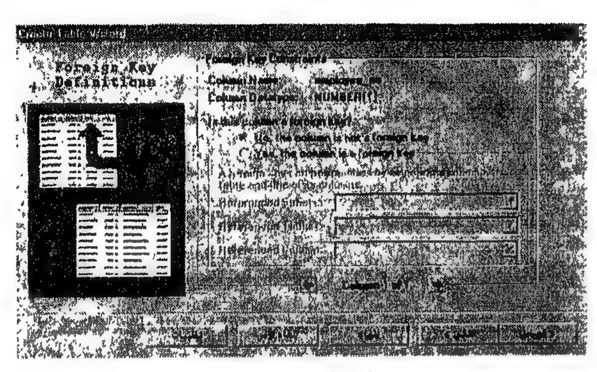
الشكل ٢٧-٩

٤- أما الصفحة الرابعة، فتسمح لك بتعريف القيدين Unique, Null على الأعمدة، انظر الشكل ٢٧-١٠



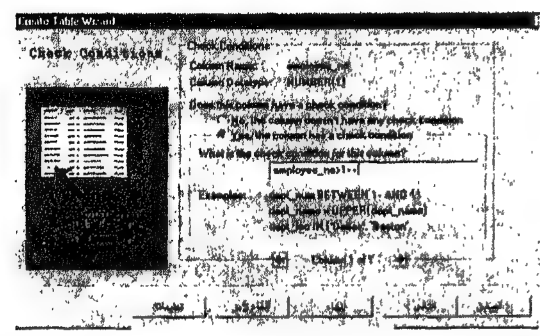
الشكل ٢٧-١٠

٥- بينما تمكنك النافذة الخامسة من تعريف المفتاح الثانوي Foreign Key على الأعمدة، انظر الشكل ٢٧-١١.



الشكل ٢٧-١١

٦- والنافذة السادسة تساعدك في تعريف شروط قيد التحقق Check constraint على الأعمدة، انظر الشكل ٢٧-١٢.

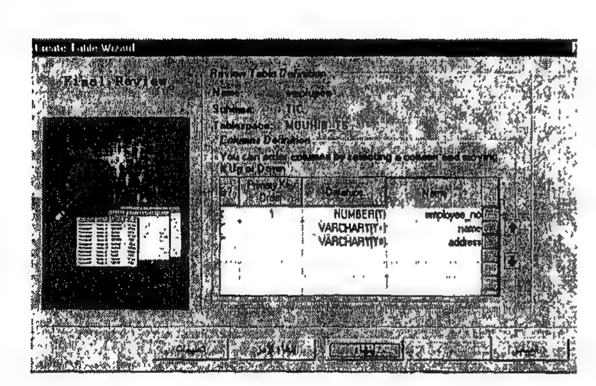


الشكل ٢٧-١٢

٧- أخيراً فإن النافذة السابعة تطلب مجموعة من المعلومات التي تساعد على حساب وسطاء التخزين والمساحة التي تم إدخالها لتعريف الجدول، انظر الشكل ٢٧ -١٣.

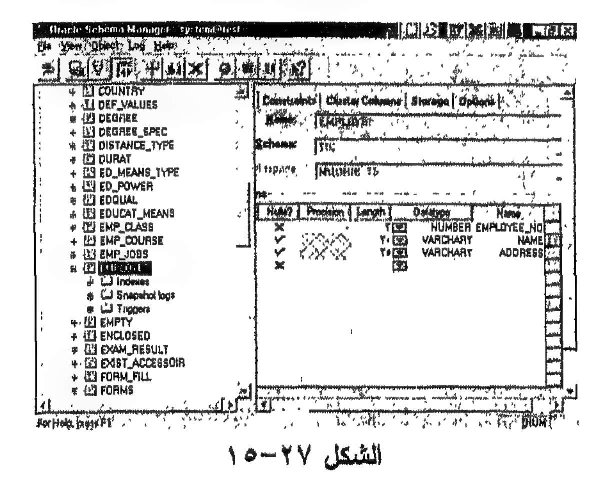
الشكل ٢٧-١٣

٨- تظهر بعد ذلك نافذة تلخيصيّة توضح البنية النهائية للجدول كما في الشكل ٢٧-١٤.



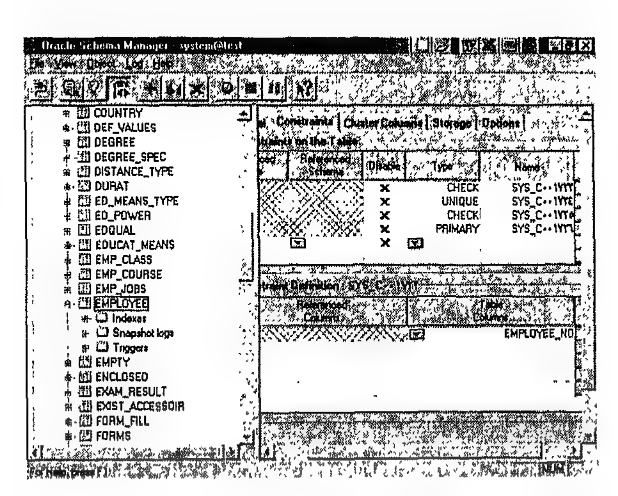
الشكل ٢٧-١٤

يمكنك ببساطة تعديل أي جدول، بالنقر على هذا الجدول، تظهر في الجزء الأيمن من النافذة المعلومات المتعلقة بهذا الجدول، يمكنك هنا إجراء التعديلات المطلوبة على هذا الجدول، حيث تستطيع تغيير توصيف حقول الجدول بالنقر على زر التبويب General (انظر الشكل ٢٧-١٥).



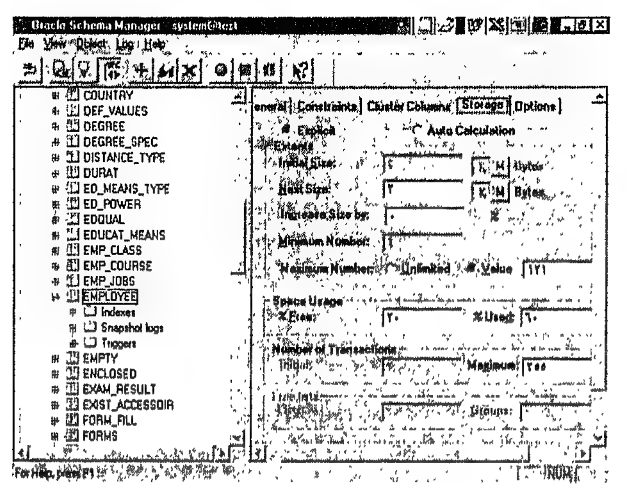
الشكل ٢٧-١٥

بالنقر على زر التبويب Constraints، يمكنك إظهار قائمة قيود التكامل على الجدول وإجراء التعديلات عليها (انظر الشكل ٢٧-١٦).



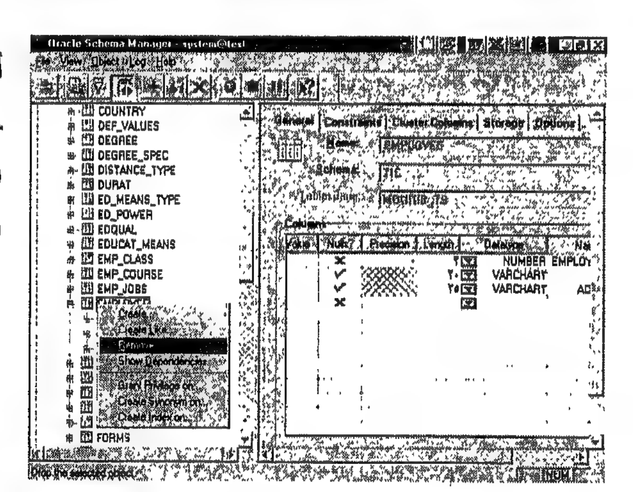
الشكل ٢٧-١٦

ولكي تستطيع تغيير قيم وسطاء التخزين أو وسطاء المساحة، يمكنك النقر على زر التبويب Storage، تظهر نافذة تشبه الشكل ٢٧-١٧، تستطيع هنا التعديل على أي من هذه القيم.



الشكل ٢٧-١٧

أخيراً إذا أردت حذف أي جدول انقر بزر الفأرة الأيمن على هذا الجدول ثم اختر Delete، انظر الشكل ٢٧-١٨.



الشكل ٢٧-١٨

الجداول المجزّأة

أتى Oracle8 بـالعديد مـن الإمكانيـات التي تتيح لك مزيداً من السيطرة على الجداول فيـزيائياً. مـن أهمها الجداول المجزأة حيث يتم تقسيم الجدول على شكل مجالات range partitioning بحيـث يـتم وضع أسطر الجداول في فضاءات جدولية مختلفة، أو تقسيم الجدول بالاعتماد على قيم المعطيات.

طبعاً بالنسبة للمستخدم فلا يختلف عليه أي شيء، لكن هناك العديد من المميزات لهذا النوع من الجداول أهمها:

- تقسـيم مقاطع المعطيات إلى أجزاء أصغر مما يؤدي إلى تصغير عملية مسح الجدول.
- تصـغير الفهارس حيث يصبح بإمكانك إنشاء فهارس منفصلة لكل جزء من الجدول، بالتالي فأن عملية الوصول إلى المعطيات تصبح أسرع بكثير.
- تسـهيل عمليات النسخ الاحتياطي والاسترداد حيث يمكن تقسيم هذه العمليات وتنفيذها بشكل متوازي لجميع أجزاء الجداول.

لكن كيف يتم إنشاء الجداول المجزأة؟

استخدم تعليمة CREATE TABLE على الشكل:

```
CREATE TABLE [ schema. ] table
(column datatype [ , column data, type] ...)
PARTITION BY RANGE (column_ list)
(PARTITION [partition] VALUE LESS THAN column value
TABLESPACE ts_ name
[,(PARTITION [partition] VALUE LESS THAN column_number
TABLESPACE ts_name])
```



```
CREATE TABLE employee،
        (id number،
```

```
        name VARCHAR2 (20),
        sal number (7,2))
PARTITION BY RANGE (sal)
(PARTITION VALUSE LESS THAN 1000 TABLESPACE t1
PARTITION VALUES LESS THAN 2000 TABLESPACE t2
PARTITION VALUSE LESS THAN 3000 TABLESPACE t3
PARTITION VALUSE LESS THAN MAXVALUE TABLESPACE t4)
```

يمكنك أيضاً إنشاء الجداول كعناصر؟

كما نعلم فإن أوراكل هو نظام إدارة قواعد معطيات علاقاتية غرضية التوجه ORDBMS Object Relational Data Base Management System لذلك يمكن تعريف الجداول كعناصر Objects بدلاً من تعريفها كأنماط معطيات DATA TYPE، بالتالي تستطيع تعريف التوابع والإجرائيات إضافةً إلى الحقول عند تعريف عنصر جدول.

```
CREATE TYPE Person_id AS OBJECT
        (id number،
        name VARCHAR2 (20),
        MEMBER FUNCTION get_id RETURN
VARACHAR2
        );
```

تستطيع الآن إنشاء جدول باستخدام عنصر الجدول الذي تم تعريفه مسبقاً مثلاً:



```
CREATE TABLE person (
        Id_per person_id،
        Addr_Per VARCHAR2 (40),
        Sal_per number (7,2));
```

تستطيع إدراج عناصر إلى هذا الجدول على الشكل التالي:



```
INSERT INTO Person VALUSE (
                Id_per (20,' AHMAD'),
                'DAMASCUS',
                3000);
```

كيفية حساب المساحة المطلوبة للجداول

سنقوم هنا بتوقع المساحة الابتدائية اللازمة لجدول في قاعدة معطيات أوراكل، وبعد ذلك حاول زيادة ١٠% مساحة إضافية لحساب حجم الكتلة التوسعية الابتدائية اللازمة للجدول.

الخطوات اللازمة لتوقع مساحة جدول هي :

١- حساب الحجم الكلي لترويسة الكتلة Header Block.

٢- حساب الحجم المتاح داخل كتلة المعطيات.

٣- حساب الحجم الوسطي لأعمدة كل سطر من الجدول.

٤- حساب الحجم المتوسط للسطر.

٥- حساب متوسط عدد الأسطر التي يمكن أن تحتويها كتلة المعطيات Data Block.

٦- حساب عدد الكتل والبايتات التي يحتاجها الجدول.

وسنفصل فيما يلي كيفية إجراء العمليات السابقة :

١- حساب الحجم الكلي لترويسة الكتلة :

Block Header = A + B
A = fixed header + variable transaction header
Fixed header = 57 bytes
*Variable transaction header = 23 * INITRANS.*
B = table directory + row directory
Table directory = 4
*Row directory = 2 * R*

حيث R هي عدد الأسطر في الكتلة (وسيتم حسابها في الخطوة 5).

→ total block header = (57 + 23) + (4 +2R) = 80 + (4 + 2R) bytes

٢- حساب الحجم المتاح داخل كتلة المعطيات :

*available data space = (block size – total block header)- ((block size – block header, part A) * (PCTFREE/100))*



يمكن معرفة قيمة الوسيط db-block-size بكتابة التعليمة :

SHOW PARAMETERS db-block-size;

لنفرض أن حجم كتلة المعطيات هو 2k وقيمة PCTFREE = 10 فإن :

available data space = (2048-(80+(4+2R))-((2048-80)(10/100)) bytes*
= (1768-2R) bytes

٣- **حساب الحجم الوسطي لأعمدة كل سطر من الجدول:** من أجل القيام بذلك يجب معرفة الأمور التالية:

- عدد الأعمدة المعرّفة ضمن الجدول.
- أنماط المعطيات المستخدمة في كل عمود.
- المساحة المتوسطة للأعمدة ذات الأطوال المتغيرة.

يمكننا استخدام تعليمة SQL التالية من أجل حساب مساحة المعطيات المجمّعة على الشكل:

```
SELECT AVG(NVL(VSIZE(col1),0) +
        AVG(NVL(VSIZE(col2),0) +
        AVG(NVL(VSIZE(coln),0)          "space of AVG row"
FROM test_table_name;
```

لنفترض أننا قمنا بإنشاء الجدول التالي :

```
CREATE TABLE t  (
A CHAR (10),
B DATE,
C NUMBER(10,2));
```

يمكن توقع مساحة المعطيات المجمعة في الجدول السابق على الشكل التالي :

- العمود a من نمط CHAR وهو بطول ثابت = ١٠ بايت (كل CHAR = ١ بايت)
- العمود b من نمط DATE وهو بطول ثابت = ٧ بايت (كل CHAR = ١ بايت)
- العمود c من نمط NUMBER وهو نمط متغير الطول، ويمكن حساب متوسط طوله على الشكل :

average length of column c = (average precision / 2 + 1)

حيث precision = ٨

= (8/2 + 1)
= 5 bytes

بالتالي : الحجم الوسطي لأعمدة كل سطر من الجدول= ١٠+٧+٥ = ٢٢ بايت

٤- **حساب الحجم المتوسط للسطر :** يمكننا حساب المساحة الصغرى التي يحتاجها كل سطر في الجداول غير التجمّعية بالمعادلة التالية :

bytes/row = row header + F +V + D

حيث row header = 3 bytes

F : تمثل الطول الكلي بالبايت لكل الأعمدة التي تخزن 255 بايتا أو أقل. عدد البايتات المطلوب لكل عمود من هذا النمط يساوي 1 بايت.

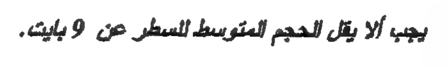
V : تمثل الطول الكلي بالبايت لكل الأعمدة التي تخزن أكثر من 250 بايتا. عدد البايتات المطلوب لكل عمود من هذا النمط يساوي 3 بايت.

D : الحجم الوسطي لأعمدة كل سطر من الجدول (من الخطوة السابقة3).

بالنسبة للجدول T الذي استخدمناه في المثال السابق :

*avg.row size,table T = (3+(1*3)+(3*0)+22) bytes*
= 28 bytes





٥- حساب متوسط عدد الأسطر التي يمكن أن تحتويها كتلة المعطيات (R):

R(avg.#of row/block) = available space/average row size

حيث : available space تم حسابها في الخطوة 2.

average row size تم حسابها في الخطوة 4.

باستكمال نفس المثال السابق نحصل على النتائج التالية :

R row/block = (1768-2)/28 bytes
⇨ *R = 58 row/block*

٦- حساب عدد الكتل والبايتات التي يحتاجها الجدول :

blocks for table = # row/R
*#bytes for table = # blocks for table * block size*



بالنسبة للجدول T، وإذا افترضنا أنه سيحتوي على 1000 سطر :

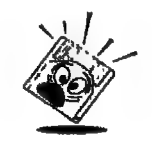
blocks for table T = 1000/58
= 17.3 blocks
*# bytes for table T = 17.3 * 2048*
= 35430 bytes (34.6KB)

نصائح هامة عند إنشاء جداول كبيرة Large Tables

يجب مراعاة بعض الأمور عند إنشاء جداول كبيرة Large Tables منها :

١. محدودية عدد المدى في الجدول والتي تتعلق بنظام التشغيل المستخدم وحجم كتلة المعطيات، فمثلا بالنسبة لأغلب قواعد البيانات التي يكون فيها حجم كتلة القاعدة مساو لـ 2k فيمكن أن تحتوي في كل مقطع على 121 مدى على الأكثر.

لذلك فإذا كان حجم الجدول سيتزايد إلى حد كبير، يجب عليك التحقق من أن الكتلة التوسعية كبيرة الحجم إلى حد كاف بحيث لا يحتوي المقطع على عدد كبير من هذه الكتل.



إذا تم حجز العدد الأعظمي من المقاطع وظلت هناك حاجة إلى مساحات إضافة، فإنه يتوجب عليك القيام بعمليات تصدير وحذف وإعادة إنشاء العناصر، ثم زيادة قيم وسطاء التخزين وأخيرا إعادة استيراد المعطيات ويمكن أن تأخذ هذه العملية أكثر من 14 ساعة من أجل الجداول الكبيرة جدا.

٢. **فصل الجداول عن الفهارس:** حاول وضع الفهارس في فضاءات جدولية منفصلة عن بقية العناصر، وإن أمكن فعلى أقراص تخزين مختلفة.

لأنك إذا احتجت إلى إجراء عملية حذف وإعادة إنشاء فهرس لجدول كبير جدا فإن الفهارس الموجودة في فضاءات جدولية مختلفة يمكنها إيجاد مساحات متجاورة Contiguous Space بشكل أسهل من الفهارس الموجودة مع بقية العناصر في نفس الفضاء الجدولي.

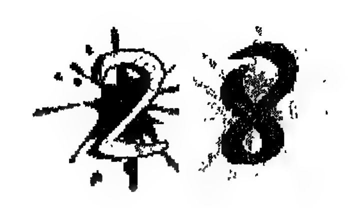
٣. **حجز أماكن تخزين مؤقتة كافية** Temporary Space.



الجداول وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للجداول هي:

- ☆ المشاهد المتعلقة بالجداول: *ALL_TABLES, USER_TABLES, DBA_TABLES*.
- ☆ المشاهد المتعلقة بأعمدة الجداول: *ALL_TAB_COLUMNS, USER_TAB_COLUMNS, DBA_TAB_COLUMNS*.
- ☆ المشاهد المتعلقة بتعليقات الجداول: *ALL_TAB_COMMENTS, USER_TAB_COMMENTS*.
- ☆ المشاهد المتعلقة بتعليقات أعمدة الجداول: *ALL_COL_COMMENTS, USER_COL_COMMENTS, DBA_COL_COMMENTS*.
- ☆ المشاهد المتعلقة بالقيود على الجداول: *DBA_CONSTRAINTS, ALL_CONSTRAINTS, USER_CONSTRAINTS*.
- ☆ المشاهد المتعلقة بالقيود على أعمدة الجداول: *DBA_CONS_COLUMNS, ALL_CONS_COLUMNS, USER_CONS_COLUMNS*.





إدارة المشاهد
Administrating Views

المشهد عبارة عن نافذة على جدول. ويمكن التعامل مع مشهد تماماً كما نتعامل مع الجداول.

لذلك فإن المشهد ليس جدولاً وإنما هو عبارة عن بنية منطقية تشبه الجدول، وهو يقوم بجلب معطياته من جداول تسمى جداول القاعدة.

تستخدم المشاهد لتسهيل الوصول إلى بعض المعطيات وإخفاء معطيات أخرى.

إدارة المشاهد باستخدام Server Manager

يمكن إجراء عمليات إنشاء وتعديل وحذف مشهد باستخدام الأداة Server Manager.

إنشاء وتعديل مشهد

يمكن إنشاء مشهد جديد باستخدام التعليمة:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
        [(alias[,alias]...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY];
```

حيث:

* **OR REPLACE**: لإعادة إنشاء المشهد إن كان موجوداً من قبل.
* **FORCE**: لإنشاء المشهد بغض النظر عن وجود أو عدم وجود جدول القاعدة.
* **NOFORCE**: لإنشاء المشهد فقط في حالة وجود جدول القاعدة.
* **view**: اسم المشهد المطلوب إنشاؤه.
* **alias**: لتحديد الرديف الموافق لعبارة استعلام المشهد.
* **subquery**: تعليمة SELECT التي تقوم بتوليد المشهد.
* **WITH CHECK OPTION**: للتحقق من أنه لايمكن إدراج أو تعديل سوى الأسطر التي يمكن الوصول إليها في المشهد.
* **constraint**: اسم القيود الموافقة للخيار السابق.
* **WITH READ ONLY**: للتحقق من أنه لايمكن إجراء أية تعليمات DML على المشهد.

إذا رغبنا بإنشاء مشهد يحتوي على المعلومات المتعلقة بموظفي القسم ٣٠ نكتب:

```
CREATE VIEW     emp30
AS SELECT       empno,ename,job
```

```
FROM          emp
WHERE         deptno=30;
```

الآن يمكنك رؤية أسطر المشهد السابق بكتابة التعليمة:

```
SELECT *
FROM emp30;
```

يمكنك الآن تعديل المشهد السابق بإضافة رديف لكل اسم عمود على الشكل:

```
CREATE OR REPLACE VIEW emp30
        (employee_no, employee_name, job_title)
AS SELECT         empno, ename, job
FROM              emp
WHERE             deptno=30;
```

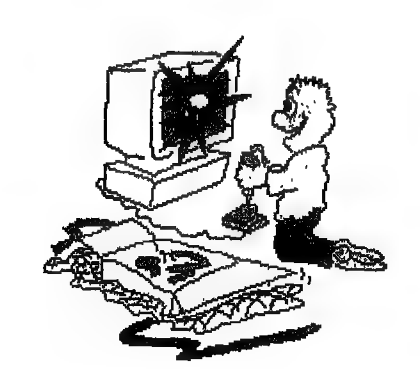
حذف مشهد

يمكن حذف مشهد تمّ إنشاؤه باستخدام التعليمة التعليمة:

```
DROP VIEW view;

DROP VIEW emp30;
```

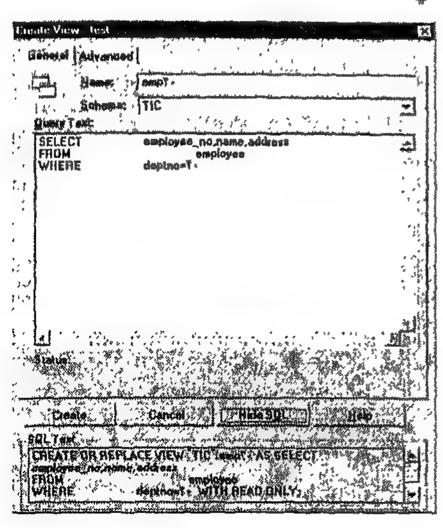




إدارة المشاهد باستخدام Storage Manager

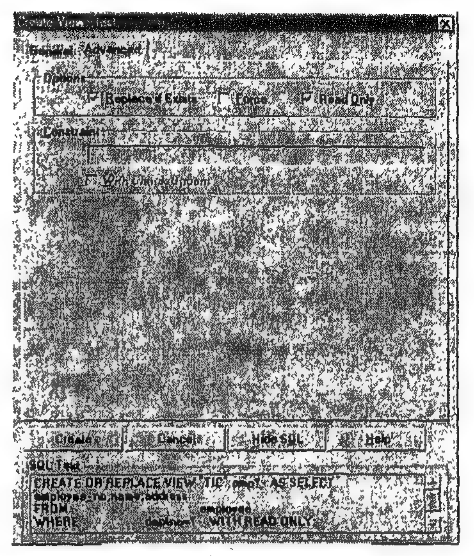
عندما تقوم بفتح نافذة Storage Manager أو نافذة Enterprise Manager، انقر على العقدة Views تظهر قائمة بالمشاهد الموجودة في قاعدة المعطيات.

- من أجل إنشاء مشهد جديد، انقر بزر الفأرة الأيمن على العقدة Views، ثم اختر Create يظهر صندوق الحوار Create View (يشبه الشكل ٢٨-١)، حدد اسم المشهد Name واسم المخطط Schema، ثم اكتب الاستفسار الموافق لمشهد ضمن Query text.

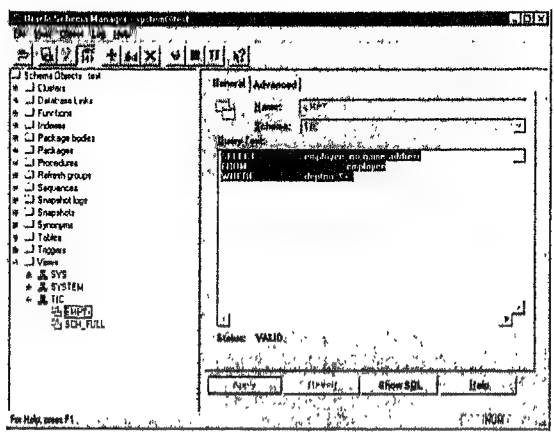


الشكل ٢٨-١

بالنقر على زر التبويب Advanced، يمكنك إضافة بعض الخيارات لإنشاء المشهد، انقر بعدها على زر Create لإنشاء المشهد (انظر الشكل ٢٨-٢). تستطيع إظهار تعليمة SQL الموافقة لعملية إنشاء مشهد بالنقر على زر Show SQL.



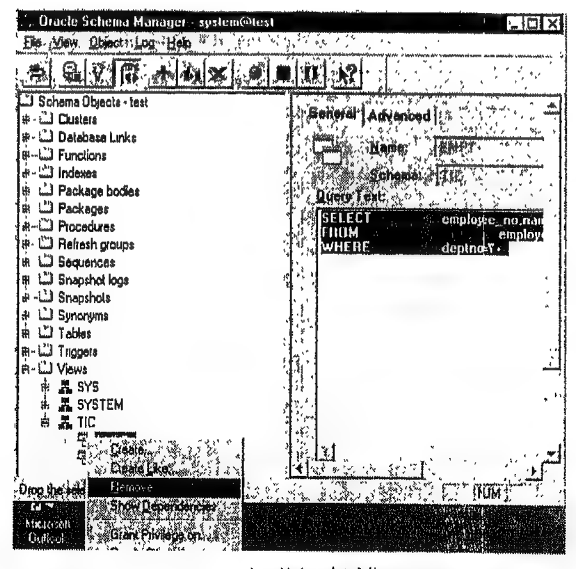
الشكل ٢٨-٢



الشكل ٢٨-٣

- تستطيع التعديل على أي مشـــهد بالـــنقر على المشهد المطلوب فتظهر المعـــلومات المتعلقة به فـــي الجـــزء الأيمن من النافذة، فتستطيع تعديل خصائص المشهد، انظر الشكل ٢٨-٣.

- لحـــذف مشهد، انقر بزر الفأرة الأيمن على المشـــهد المطـــلوب حذفه ثم اختر Remove (انظر الشكل ٢٨-٤).



الشكل ٢٨-٤

المشاهد وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للمشاهد هي:

- المشاهد المتعلقة بالمشاهد: *ALL_VIEWS, USER_VIEWS, DBA_VIWS*.
- المشاهد المتعلقة بأعمدة المشاهد: *ALL_TAB_COLUMNS, USER_TAB_COLUMNS, DBA_TAB_COLUMNS*.
- المشاهد المتعلقة بتعليقات المشاهد: *ALL_TAB_COMMENTS, USER_TAB_COMMENTS*.
- المشاهد المتعلقة بتعليقات أعمدة المشاهد: *ALL_COL_COMMENTS, USER_COL_COMMENTS, DBA_COL_COMMENTS*.





إدارة المرادفات
Administrating Synonyms

المرادفات عــبارة عن عناصر قاعدة معطيات تدل على عناصر أخرى في النظام.

فهــي لذلــك أســماء رديفة للجداول أو المشاهد أو السلاسل أو الوحدات البرمجية.

تستخدم المرادفات لإخفاء بعض التفاصيل عن المستخدمين والمخطط الذي يحتوي العنصر أو موقع هذا العنصر.

يمكن للمرادفات أن تكون بإحدى الحالتين: عامة Public، أو خاصة Private.

فالمرادفات العامة Public synonyms يمتلكها مخطط العناصر PUBLIC وتكون متاحة لكل مستخدم في قاعدة المعطيات.

أمــا المرادفات الخاصة Private synonyms فتكون عادةً موجودة في مخطط المستخدم الذي قام بإنشائها وهو الوحيد الذي يستطيع التحكم بمن يستطيع الوصول إليها.

إدارة المرادفات باستخدام Server Manger



تستطيع مثلاً استخدام تعليمة SELECT على الشكل:

*SELECT * FROM moh;*

بدلاً من:

*SELECT * FROM mohib.employee;*



ويمكن ببساطة إنشاء مرادف باستخدام تعليمة *CREATE SYNONYM*

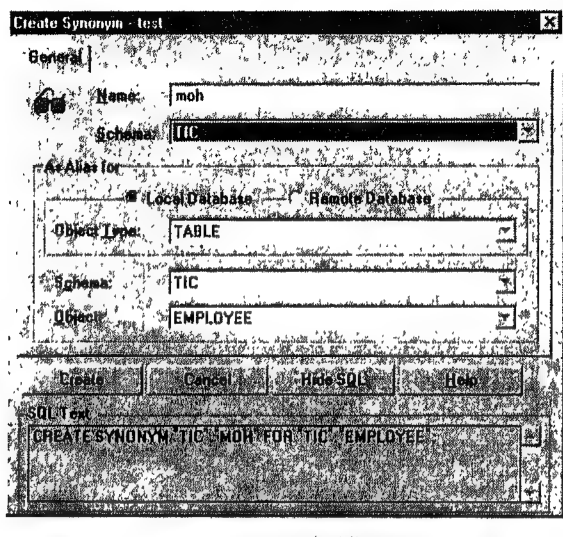
مثلاً:

CREATE PUBLIC SYNONYM MOH for mohib.employee;



إدارة المرادفات باستخدام Enterprise Manager

يمكن التعامل مع المرادفات من خلال الأداة Schema Manager أو الأداة Enterprise Manager، حيث يمكن إنشاء مرادفاً جديداً بالنقر بزر الفأرة الأيمن على عقدة Synonyms ثم طلب الخيار Create. يظهر صندوق الحوار Create Synonyms انظر الشكل ٢٩-١.

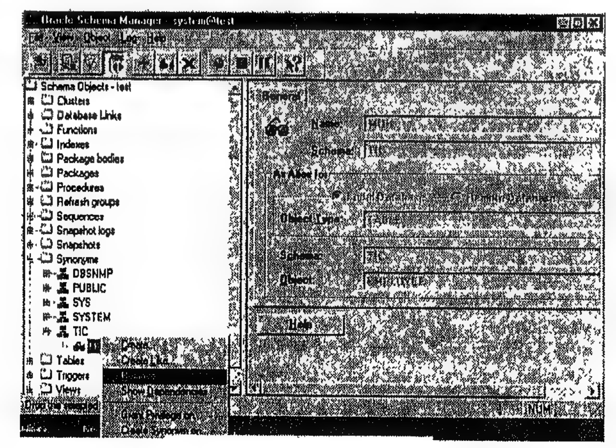


الشكل ٢٩-١

اكتب اسم المرادف Name ومخطط العناصر Schema، وحدّد إن كان رديف لقاعدة محلية Local database أو لقاعدة بعيدة Remote Database. حدد كذلك نمط العنصر الذي سيدل عليه المرادف Object type، والمخطط الذي يتواجد فيه هذا العنصر Schema واسم هذا العنصر Object، ثم انقر بعدها على زر Create لإنشاء المرادف.

تستطيع بالنقر على زر Show SQL إظهار تعليمة SQL الموافقة لإنشاء المرادف.

تستطيع حذف أي مرادف بالنقر بزر الفأرة الأيمن عليه واختيار Remove (انظر الشكل ٢٩-٢).

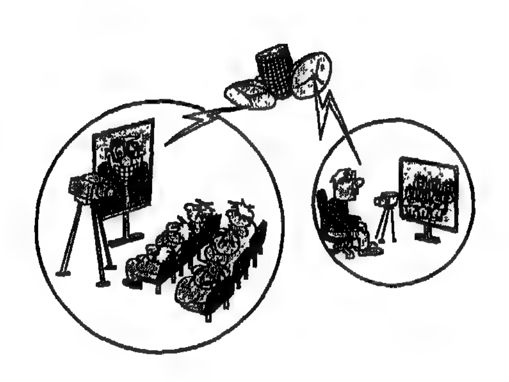


الشكل ٢٩-٢

المرادفات وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للمرادفات هي:

☆ المشاهد المتعلقة بالمرادفات: USER_SYNONYMS, DBA_SYNONYMS, ALL_SYNONYMS.







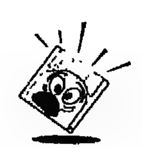
إدارة الفهارس
Administrating Indexes

الفهارس عــبارة عــن بــنى منطقية ترتبط بالجداول أو التجمعات. وهي تســتخدم لتســريع تــنفيذ تعليمات SQL. ويمكن إنشاؤها بشكل منفصل.

لذلــك فــإن الفهارس تساعد في الوصول وبشكل أسرع إلى المعلومات، لأنها تدل مباشرةً على مواقع الأسطر التي تحتوي المعلومات المطلوب البحث عنها.

كما أن الفهارس غير متعلقة فيزيائياً أو منطقياً بالمعطيات الموجودة في الجداول المرتبطة، لذلك يمكن إنشاؤها أو حذفها في أي وقت دون التأثير على هذه الجداول أو على الفهارس الأخرى.

بعــد أن يــتم إنشــاء فهــرس، يمكن بشكل تلقائي استخدامه من قبل أوراكل، فعند إجراء الــتغييرات على المعطيات كإضافة أو حذف أو تعديل الأسطر، فإنه سيؤثر تلقائياً في كل الفهارس الموافقة.



قد تساعد الفهارس في تحسين أداء سرعة الحصول على المعلومات، لكن في حال إنشاء الكـثير مـن الفهارس على جدول ما، فإن هذا بالتأكيد سيقلل من أداء النظام وسيبطئ عمليات الإضافة أو الحذف أو التعديل على أسطر الجدول.

يمكـن للفهارس أن تكون وحيدة Unique أي أنها تضمن عدم وجود سطرين في الجدول يحـتويان عـلى قيم متكررة في الأعمدة التي عرّفت عليها الفهارس، كما يمكن ألا تكون وحيدة Non Unique.

كذلك يمكن إنشاء فهرساً مركباً Composite Index على عدة أعمدة في جدول.

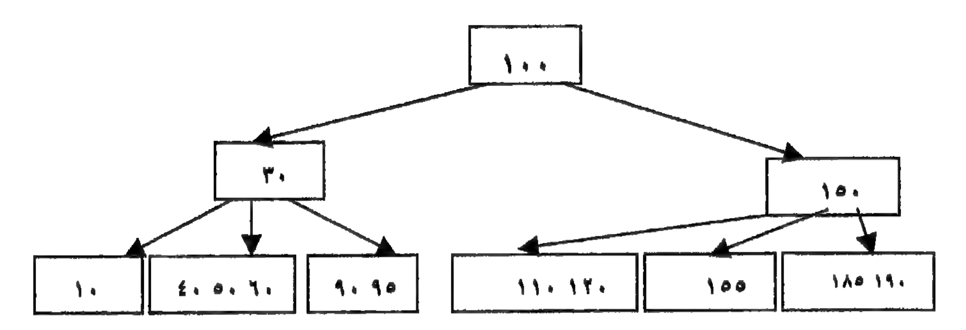
وعندما يتم إنشاء فهرس، فإنه يتم تلقائياً إنشاء شريحة فهرسة Index Segment لاحتواء معطيـات الفهـرس، حيث يمكن التحكم بعملية حجز مساحات لشريحة الفهرسة بتحديد قيم وسطاء التخزين الخاصة بشريحة الفهرسة.

البـنية الداخـلية لـلفهارس ضـمن أوراكل Internal Structure of Indexes

يسـتخدم أوراكـل الفهارس من نمط B*-tree وهي عبارة عن أشجار متوازنة من أجل موازنة عمليات الوصول إلى أي سطر. ويمتلك هذا النوع من البنى الميزات التالية:

- جميع الأوراق في الشجرة لها نفس المستوى، لذلك فإنّ جلب أي سجل من أي مكان في الفهرس يأخذ تقريباً نفس الزمن.
- تبقى الفهارس B*-tree متوازنة بشكل تلقائي دائماً.
- جميـع الصـفحات فـي شجرة B*-tree تمتلئ حتى ثلاثة أرباعها كمعدل وسطي.
- يزودنا هذا النوع من الأشجار بطريقة ممتازة جداً في جلب المعطيات بالنسبة لعدد كبير من الاستفسارات.
- عمليات الإدراج والتعديل والحذف على هذه الأشجار فعالة دوماً.

- أداء هـــذا الـــنوع مـــن الأشجار جيد بالنسبة للجداول الصغيرة والكبيرة على السواء، ولا يتأثر عند زيادة حجوم الجداول.
- يوضح الشكل ٣٠-١ شجرة من نمط B*-tree:



الشكل ٣٠-١

لأي شـــجرة معمعـــة درجة n بحيث لا يمكن لأي صفحة من صفحات الشجرة أن تحتوي على عناصر أقل من n أو أكثر من ٢n.
عـــدا الصـــفحة الجـــذر التي يمكن أن تحتوي على عنصر واحد فقط . ففي شجرة الشكل السابق فإن درجة الشجرة هي ٢، لذلك فإن عدد عناصر كل صفحة لا يمكن أن يقل عن ٢ ولا يتجاوز ٤.

كيف يمكن الاستفادة من ميزات الفهارس؟

بعد فهرسة عمود أو أكثر من أعمدة الجدول، فإنه يصبح من الضروري تضمين الأعمدة المفهرسة في عبارة WHERE ضمن تعليمة SQL لأن المحلل optimizer سيقوم مباشرةً بمسح الفهرس بدلاً من الجدول للوصول إلى المعطيات المطلوبة مما سيؤدي إلى تسريع عملية الوصول إلى هذه المعطيات بشكل كبير.

لكن هناك معايير للفهرسة يجب اتباعها؟

يوجد العديد من المعايير التي تساعدك في اتخاذ القرار بفهرسة أو عدم فهرسة الأعمدة وهي:

- قم بفهرسة الجداول عندما يختار الاستعلام عدداً قليلاً من أسطر هذه الجداول، لأن الاستعلامات تعطينا عدداً كبيراً من الأسطر فتلغي الهدف من الفهارس. لذلك استخدم الفهارس عندما تعطي الاستعلامات أقل من ٥٪ من الأسطر في الجدول.
- حاول ألا تقوم بفهرسة الجداول التي تجري عليها عمليات الإدراج والتعديل والحذف بشكل متكرر، لأن عملية الفهرسة هنا ستبطئ كثيراً من العمليات السابقة على الجداول.
- حاول عدم إنشاء فهارس على أعمدة تحتوي على قيم مكررة كثيراً، فمثلاً الأعمدة التي تأخذ القيم False أو True غير مناسبة للفهرسة.
- حاول فهرسة الجداول التي يتم إجراء استعلامات بسيطة عليها باستخدام عبارات WHERE بسيطة، أما عبارات WHERE المعقدة فقد لا تستفيد من الميزات الموجودة في الفهارس.

إدارة الفهارس باستخدام Server Manager

إنشاء فهرس جديد

يمكن إنشاء فهرس باستخدام تعليمة CREATE INDEX على الشكل:

CREATE [UNIQUE|BITMAP] INDEX [schema.]index
ON [schema.]table (column, ...) |
ON CLUSTER [schema.]cluster;
[INITRANS integer]
[MAXTRANS integer]
[TABLESPACE tablespace]
[STORAGE storage_clause]
[PCTFREE integer]
[NOSORT]
[RECOVERABLE|UNRECOVERABLE]
[PARALLEL parallel_clause]

حيث:

- **UNIQUE**: لتحديد أن قيمة العمود المفهرس وحيدة.
- **BITMAP**: لتحديد نمط الفهرس النقطي.
- **schema**: اسم مخطط العناصر.
- **table**: اسم الجدول الذي سيتم إنشاء الفهرس عليه.
- **column**: اسم العمود.
- **CLUSTER**: اسم التجمّع الذي سيتم إنشاء الفهرس عليه.
- **INITRANS**: لتحديد عدد مداخل التحويلات التي سيتم حجزها مسبقاً لكل ترويسة كتلة في المقطع. القيمة الافتراضية هي ٢ وطول كل مدخل ٢٣ بايت.
- **MAXTRANS**: لـتحديد عـدد الـتحويلات التي يمكنها الوصول بشكل متزامن إلى الكتلة، القيمة الافتراضية تساوي ٢٥٥.
- **TABLESPACE**: اسم الفضاء الجدولي الذي سيتم إنشاء الفهرس فيه.
- **STORAGE**: لتحديد وسطاء تخزين الفهرس.
- **PCTFREE**: المساحة المحجوزة لمداخل فهارس إضافية.

- ❋ **NOSORT**: لإخبار مخدّم أوراكل بأن أسطر الجدول قد تمّ فرزها تصاعدياً من قبل.
- ❋ **RECOVERABLE**: لــتحديد أن إنشاء الفهرس سيتم تسجيله في ملفات الإرجاع، أما الخيار المعاكس فهو UNRECOVERABLE.
- ❋ **PARALLEL**: لإنشاء الفهرس بشكلٍ متوازٍ.



```
CREATE INDEX ind_emp_empno ON EMP(EMPNO)
TABLESPACE ts_mohib
STORAGE      (INITIAL 100K
              NEXT 100K
              PCTINCREASE 50);
```

تعديل فهرس

يمكن تعديل فهرس باستخدام التعليمة:

```
ALTER INDEX [schema.]index
[PCTFREE integer]
[INITRANS integer]
[MAXTRANS integer]
[STORAGE storage_clause]
[ALLOCATE EXTENT          ([SIZE integer [K|M]]
                           [DATAFILE 'filename']
                           [INSTANCE integer], ...)
[DEALLOCATE UNUSED [KEEP integer [K|M]]]
[REBUILD      [PARALLEL integer | NOPARALLEL]
              [RECOVERABLE|UNRECOVERABLE]
              [TABLESPACE tablespace]];
```

حيث:

- ❋ **SIZE**: لتحديد حجم المدى بالبايت.
- ❋ **DATAFILE**: لــتحديد ملف المعطيات في فضاء الفهرس الذي سيحتوي المدى الجديد.
- ❋ **INSTANCE**: لجعــل المــدى الجديد متاحاً للهيئة الجديدة، ويستخدم هذا الخيار فقط مع نسخة Oracle٧ مع خيار التوازي.

* DEALLOCATE UNUSED: لإلغـــاء حجز المساحة غير المستخدمة في نهاية الجدول بشكل خارجي، وجعل المساحة الفارغة متاحة للمقاطع الأخرى.
* KEEP: لتحديد عدد البايتات التي سيمتلكها الجدول بعد إلغاء الحجز.
* PARALLEL: لاســتخدام الإجـــراء المتوازي عند بناء الفهرس، والخيار المعاكس هو NOPARALLEL.



ALTER INDEX ind_emo_emono
MAXTRANS 10;

حذف فهرس

لحذف فهرس نستخدم التعليمة:

DROP INDEX [schema.]index;



DROP INDEX ind_emp_empno;



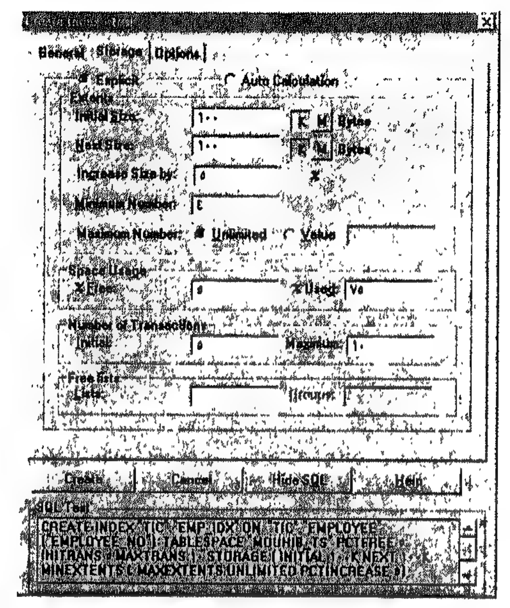
إدارة الفهارس باستخدام Schema Manager

عند فتح نافذة Schema Manager أو Enterprise Manager يظهر مخطط عناصر قاعدة المعطيات، وتظهر العقدة Index. تستطيع هنا إنشاء فهرس جديد بالنقر بزر الفأرة الأيمن ثم اختر Create يظهر صندوق الحوار Create Index، كما في الشكل ٣٠-٢.

الشكل ٣٠-٢

حدّد في صندوق الحوار السابق اسم الفهرس Name والمخطط الذي سيتم فيه وضع الفهرس Schema ونمط الفهرس Index Type. كذلك حدّد هل سترغب بفهرسة جدول Table أو تجمّع Cluster، والمخطط الذي يتواجد فيه هذا الجدول أو المجمع Schema، حدّد أيضاً اسم الجدول Table، وحدّد أعمدة الفهرس وترتيب هذه الأعمدة Order. أخيراً حدّد نوع الفهرس Unique أو Sorted أو recoverable.

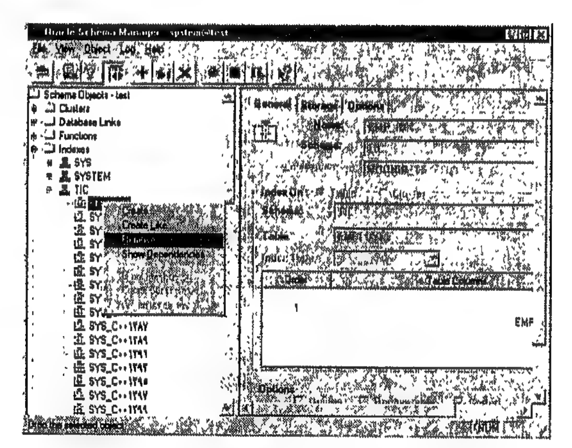
يمكـــنك بالـــنقر عـــلى زر التبويب Storage التحكّم بقيم وسطاء تخزين ووســـطاء مســـاحة الفهرس، انظر الشكل ٣٠-٣، تستطيع رؤية تعليمة SQL الموافقة لإنشاء الفهرس بالنقر على زر SHOW SQL.



الشكل ٣٠-٣

تستطيع التعديل على فهرس بالنقر عليه ضمن عقدة Index وإجراء الـــتعديلات مباشـــرة فـــي نـــافذة خصائص هذا الفهرس.

أما لحذف فهرس فننقر عليه بزر الفأرة الأيمن ثم نختار Remove (انظر الشكل ٣٠-٤).



الشكل ٣٠-٤

الفهارس المجزأة Partitioned Indexes

كمـــا في الجداول، فإن الفهارس يمكن أن تكون مجزأة، مع الأخذ بعين الاعتبار أنه قد لا تكون الجداول الموافقة مجزأة.

يوجــد نوعـــان مختلفان من الفهارس المجزأة المتاحة ضمن Oracle٨: الفهارس العامة Global Indexes والفهارس المحلية Local Indexes.

الفهارس العامة Global Indexes

يمكـــن اعتبار هذا النوع من الفهارس كفهرس B*- tree وحيداً على كامل الجدول، وهو يحتوي على معلومات عن كل الأسطر في جميع أجزاء الجدول.

```
CREATE INDEX ''MOH'', Person_isc1
ON Person (id)
PARTITION BY RANGE (id)
PARTITION pt1 VALUESLESS THAN (100) TABLESPACE
ts1;
PARTITION pt2 VALUES LESS THAN (MAXVALUE) TABLESPACE
ts 2 );
```

الفهارس المحلية Local Indexes

بعكـــس الفهارس العامة، فإن الفهارس المجزأة المحلية يتم إنشاؤها بشكل منفصل على كل جزء.



```
CREATE INDEX ''MOH'', Person_ise2
UN Person (id)
LOCAL;
```

الفهارس النقطية Bitmap Indexes

يســتخدم هذا النوع من الفهارس مع الأعمدة التي تأخذ قيماً متعددةً وقليلةً، حيث يتم وضع bitmap لمحدّد السطر ROWID، وهو يدل على الأسطر الموافقة لعنصر الفهرس. فإذا كان البت محدد (قيمته ١) فهذا يدل على أن السطر الموافق يحتوي على قيمة المفتاح، أما إذا كان البت غير محدد (قيمته ٠) فهذا يدلّ على أنّ السطر الموافق لا يحتوي على قيمة المفتاح.

لنأخذ المثال التالي:

ID	Name	Sex
1	MOHIB	M
2	LAMIS	F
3	DYA	M
4	SAMER	M
5	ASMA	F

M	F
1	0
0	1
1	0
1	0
0	1

Bitmap on Index Sex

يســتخدم هذا النوع من الفهارس على الحقول التي لها تكرار قيم منخفض كالحالة العائلية ومستوى الدخل وغيرها.

من أجل إنشاء فهرس نقطي نستخدم تعليمة CREATE BITMAP INDEX مثلاً:

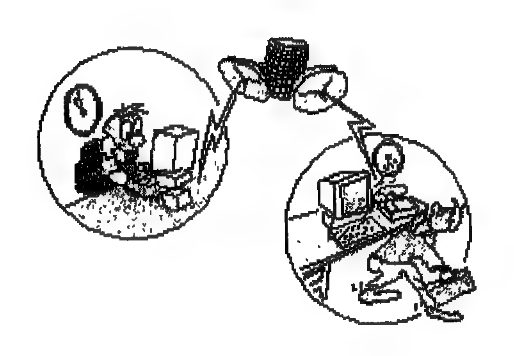


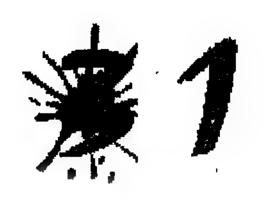
```
CREATE BITMAP INDEX '' MOH ''.Person_ ix3
ON PERSON (SEX);
```

الفهارس وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للفهارس هي:

- ☆ المشاهد المتعلقة بالفهارس: *ALL_INDEXES, USER_INDEXES, DBA_INDEXES*
- ☆ المشاهد المتعلقة بالأعمدة المفاتيح للفهرس على الجداول والتجمّعات: *ALL_IND_COLUMNS, USER_IND_COLUMNS, DBA_IND_COLUMNS*.







إدارة السلاسل
Administrating Sequences

تحتاج في كثير من الأحيان إلى توليد سلسلة أرقام، وذلك لاستخدامها في قاعدة معطيـــاتك، فقد ترغب مثلاً باستخدام هذه الأرقام في التعريف عن سجل خاص.

يقـــوم أوراكل بتزويدك بمولد سلاسل Sequence Generator يستطيع توليد سلسلة أرقام يمكن أن يصل طولها حتى ٣٨ خانة دون الحاجة إلى قفل السجلات يدوياً.

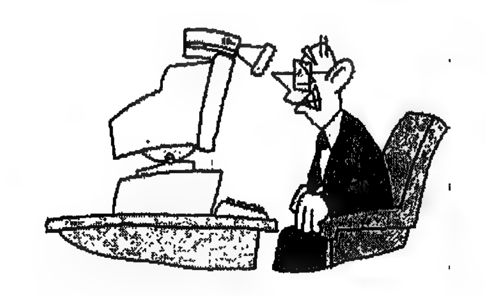
إدارة السلاسل باستخدام Server Manager

يمكنك استخدام تعليمة CREATE SEQUENCE لإنشاء سلسلة جديدة، حيث تأخذ هذه التعليمة الشكل:

```
CREATE SEQUENCE sequence
[INCREMENT BY integer]
[START WITH integer]
[MAXVALUE integer or NOMAXVALUE]
[MINVALUE integer or NOMINVALUE]
[CYCLE or NOCYCLE]
[CACHE integer or NOCACH]
[ORDER or NOORDER]
```

```
CREATE SEQUENCE seq1
INCREMENT BY 1
START WITH 1
MAXVALUE 100
NOCACH
NOCYCLE
```





استخدام السلاسل

يمكنك توليد قيمة جديدة ضمن سلسلة باستخدام العامل NEXTVAL على الشكل:

sequence_name.NEXTVAL

أما لإعادة استخدام القيمة الحالية ضمن سلسلة فنحتاج المعامل CURVAL الذي يستخدم على الشكل:

sequence_name.CURVAL



```
INSERT INTO dept (deptno, name, loc)
VALUES (seq1.NEXTVAL, 'MARKETING', 'DAMAS');
```

يمكنك معرفة القيمة الحالية للسلسلة seq١ باستخدام التعليمة:

```
SELECT seq1.CVRVAL FROM dual;
```



تتزايد السلاسل عند كل عملية وصول إليها بشكل مستقل عن إجراء COMMIT أو ROLLBACK. لكن في حال قام تحويل بتوليد سلسلة ثم تمّ التراجع عن ذلك، فإنه لا يتم استبدال قيمة السلسلة مما يؤدي إلى حدوث انقطاع في قيم السلسلة.

تعديل سلسلة ALTER SEQUENCE

يمكنك استخدام تعليمة ALTER SEQUENCE لتعديل سلسلة وذلك على الشكل:

```
ALTER SEQUENCE sequence
[INCREMENT BY integer]
[START WITH integer]
[MAXVALUE integer or NOMAXVALUE]
[MINVALUE integer or NOMINVALUE]
[CYCLE or NOCYCLE]
[CACHE integer or NOCACH]
[ORDER or NOORDER]
```



```
ALTER SEQUENCE seq1
INCREMENT BY 1
MAXVALUES 9999
NOCACH
NOCYCLE
```

حذف سلسلة Drop Sequence

تستطيع حذف سلسلة باستخدام التعليمة:

```
DROP SEQUENCE sequence
```

```
DROP SEQUENCE seq1;
```





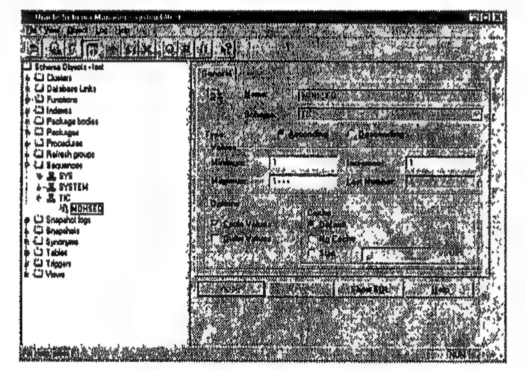
إدارة السلاسل باستخدام الأداة Schema Manager

لإنشــاء سلسلة جديدة ضمن Schema Manager، انقر بزر الفأرة الأيمن على العقدة Sequence ثــم اخــتر Create، يظهــر صــندوق الحوار Create Sequence ، انظر الشكل ٣١-١.

تســتطيع من خلال صندوق الحــوار هــذا تحديــد اسم السلســلة Name والمخطط الــذي ســيتم إنشــاءها فيه Schema ونمــط السلســلة تصــاعدية Ascending أو تنازلية Descending، يمكــنك أيضاً تحديد القيمــة الدنيــا Minimum والقيمة العــليا Maximum وتــزايد هــذه السلســـلة Increment، والقيمـــة الابــتدائية لهــذه السلســلة Initial. تســتطيع كذلــك تحديد فيما إذا كانت هــذه السلسلة دوارة Cycle أو غير دوارة No Cycle.

الشكل ٣١-١



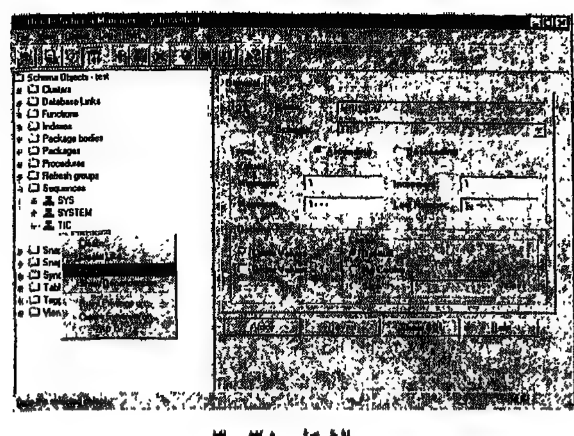


الشكل ٣١-٢

أخيراً يمكنك تحديد إن كان سيتم وضع أرقام السلسلة في ذاكرة مخبئية Cache أم لا No Cache وعدد مداخل هذه الأرقام size.

يمكنك تعديل أي سلسلة بالنقر على أيقونة هذه السلسلة، تظهر في الجزء الأيمن الحقول التي يمكنك التعديل عليها (انظر الشكل ٣١-٢).

يمكنك من خلال هذه النافذة حذف سلسلة أرقام بالنقر بزر الفأرة الأيمن على أيقونة هذه السلسلة واختيار الأمر Remove (انظر الشكل ٣١-٣).

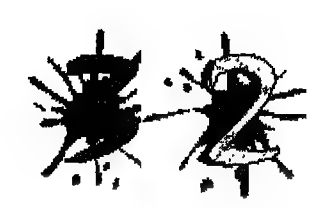


الشكل ٣١-٣

السلاسل وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للفهارس هي:

☆ المشاهد المتعلقة بالسلاسل: ALL_SEQUENCES, USER_SEQUENCES, DBA_SEQUENCES.





إدارة التجمّعات

Administrating Clusters

وهي عـبارة عـن طريقة اختيارية لتخزين الجداول في قاعدة المعطيات ويمكنها زيادة أداء عمليات الإدخال والإخراج وتقليص حجم التخزين.

يمكن للجداول التي تتشارك بأعمدة معينة أن تتجمّع حول هذه الأعمدة وهذا يؤدي إلى تسريع عملية الوصول إلى أسطر هذه الجداول.

لاحظ أن التجمّعات تؤثر في كيفية تخزين المعطيات فقط.

عـندما يقـوم أوراكـل بجلب المعطيات من القرص، تتم هذه العملية على كتل المعطيات لا عـلى الأسطر ، لذلك إذا تم تخزين المعطيات سويّة فإنه سيتم نسخها من قرص التخزين إلى الذاكـرة سوية في نفس كتلة المعطيات. وعندما تتم قراءة كتلة المعطيات، تقرأ جميع المعطيات الخاصة بالجداول التجمعيّة الموجودة في كتلة المعطيات هذه، وهذا يعطينا فائدة حقيقية.

لذلك إذا كان لديك جدولان يحتويان على معطيات مرتبطة ويتم الوصول إليهما سويةً بشكل مـتكرر، يفضّل استخدام التجمّعات من أجل تحسين الأداء وذلك عن طريق الشحن المسبق للمعطيات المرتبطة في SGA.

لنفرض مثلاً أنّ لدينا جدولين DEPT, EMP المتشاركين بالحقل DEPTNO.

ففي حال عدم تجميعهما فإنّ كل جدول سيتم تخزينه بشكل منفصل عن الآخر وبالتالي فإن المعطيــات المرتــبطة ســيتم تخزينها منفصلة مما يؤدي إلى استخدام مساحة تخزين أكبر (انظر الشكل ٣٢-١).

EMP Table

EMPNO	DNAME	DEPTNO
932	ضياء كرّاز	20
1000	سامر سعيد	10
1139	مهيب النقري	20
1277	لميس فرحة	20
1321	أسما قصقوص	10
1841	عامر سعد	10

DEPT Table

DEPTNO	DNAME	LOC
10	مبيعات	دمشق
20	توزيع	حلب

الشكل ٣٢-١

أمـــا في حال تجميع هذين الجدولين بالنسبة للعمود DEPTNO، فإنه سيتم تخزينهما فيزيائياً في نفس كتل المعطيات (انظر الشكل ٣٢-٢).

الشكل ٣٢-٢

وعلى الرغم من أن التجمّعات تحسّن من أداء الاستفسارات، إلا أنها قد تقلّص أداء عمليات إدراج وحذف وتعديل السجلات. لذلك يفضّل تجميع الجداول التي تحقّق الشروط التالية:

- يتم الاستفسار عنها بشكل متكرر.
- لا تجري عليها التعديلات كثيراً.
- تكون عادة جداول مندمجة Joined table.

ويكـــون مفتاح التجمّع Cluster key هو العمود المشترك بين الجداول التجمّعية، ويفضّل أن يحتوي هذا العمود على قيم مختلفة إلى حدّ كبير.



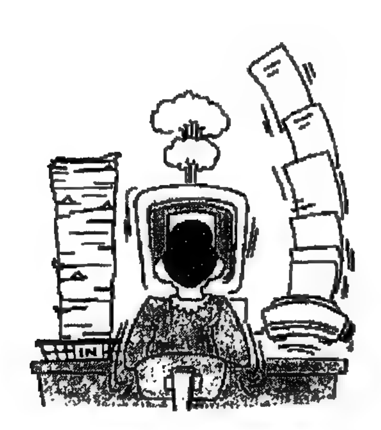
لا يمكن أن يحتوي مفتاح التجميع على أعمدة من نمط LONG أو LONG RAW.



يحــتاج تخــزين الجداول المتجمعة إلى كتل معطيات أكثر من تخزين كل جدول لوحده، لأن كتل المعطيات تكون مشتركة في التجمّع.

أمــا فهرس التجمّع Cluster index فهو عبارة عن فهرس يتم تعريفه على أعمدة مفتاح التجمّع، وهو يستخدم للبحث عن قيم مفتاح التجمّع بشكل أسرع كما أنها تؤثر مباشرة على كــتل المعطيات التي تحتوي قيم هذا المفتاح وتسمح أيضاً بالوصول إلى الأسطر بأقل عدد من عمليات في الإدخال والإخراج I/O.

ويمكن تخزين التجمّع وفهارس هذا التجمّع في فضاءات جدولية مختلفة.



إدارة التجمّعات باستخدام الأداة Server Manager

إنشاء تجمّع جديد Create Cluster

يمكن إنشاء تجمّع جديد باستخدام تعليمةSQL التالية:

```
CREATE CLUSTER [schema.] cluster (column datatype, ...)
[PCTFREE integer]
[PCTUSED integer]
[INITRANS integer]
[MAXTRANS integer]
[SIZE integer [K|M]]
[TABLESPACE tablespace]
[STORAGE storage_clause]
[INDEX]
```

حيث:

☆ **SIZE**: الحجم اللازم لتخزين جميع أسطر قيمة مفتاح تجمّع محددة.

☆ **INDEX**: لإنشاء فهرس التجمّع.

ويمكن إنشاء فهرس التجمّع بشكل منفصل عن إنشاء التجمّع نفسه باستخدام التعليمة:

```
CREATE INDEX [schema.]index
ON CLUSTER [schema.]cluster
[INITRANS integer]
[MAXTARNS integer]
[TABLESPACE tablespace]
[STORAGE storage_clause]
[PCTFREE integer]
[NOSORT]
[RECOVERABLE|UNRECOVERABLE]
```



سنقوم بإنشاء تجمّع بالاسم CLUSTER_T١_T٢ مع العمود DEPTNO كمفتاح للتجمّع.

سنضع التجمّع في الفضاء الجدولي TBS_DATA ونحدد مساحة ٤٠٠ bytes لتخزين الأسطر المرتبطة.

```
CREATE CLUSTER Cluster_T1_T2  (deptno Number(3))
SIZE 400
```

```
TABLESPACE tbs_data
STORAGE (INITIAL 30K);

CREATE INDEX I_Clu_T1_T2
ON CLUSTER cluster_T1_T2
TABLESPACE tbs_index;

CREATE TABLE T1 (name VARCHAR2 (20), hiredate DATE, deptno
number (3))
CLUSTER cluster_T1_T2 (deptno);

CREATE TABLE T2 (deptno number (3), deptname VARCHAR2 (15) )
CLUSTER cluster_T1_T2 (deptno);
```

تعديل التجمّعات Altering Clusters

يمكن إجراء التعديل على تجمّع باستخدام التعليمة:

```
ALTER CLUSTER [schema.] cluster (column datatype, ...)
[PCTFREE integer]
[PCTUSED integer]
[SIZE integer [K|M]]
[INITRANS integer]
[MAXTRANS integer]
[STORAGE storage_clause]
[ALLOCATE EXTENT        ([SIZE integer [K|M]
                        [DATAFILE 'filename'
                        [INSTANCE integer]);
```

حيث:

- ☆ **ALLOCATE EXTENT**: لحجز مدى جديد.
- ☆ **SIZE**: لتحديد حجم المدى الجديد.
- ☆ **DATAFILE**: اسم ملف المعطيات المرتبط بالفضاء الجدولي الخاص بالتجمّع.
- ☆ **INSTANCE**: يستخدم في الخيار المتوازي لتحديد رقم الهيئة التي سيتم استخدامها.

```
ALTER CLUSTER Cluster_T1_T2
STORAGE (NEXT 200K PCTINCREASE 30);
```



حذف التجمّعات Dropping Clusters

يمكنك حذف التجمّعات عندما لا تصبح بحاجة إلى جداول هذه التجمّعات.

وعندما تقوم بحذف تجمّع:

- تحذف جميع جداول التجمّع.
- يحذف فهرس التجمّع.
- يتم إرجاع جميع كتل المدى إلى الفضاء الجدولي.

تأخذ تعليمة حذف التجمّعات الشكل:

```
DROP CLUSTER [schema.]cluster
[INCLUDING TABLES [CASCADE CONSTRAINTS]]
```

حيث:

- **INCLUDING TABLES**: تقوم بحذف جميع الجداول التي تنتمي إلى التجمّع، وإذا لم يحدّد هذا الخيار فيجب حذف الجداول أولاً وإلا فستظهر رسالة خطأ ولا يتم حذف التجمّع.
- **CASCADE CONSTRAINTS**: لحذف جميع قيود التكامل المرجعي المرتبطة. وإذا لم يحدد هذا الخيار، ستظهر رسالة خطأ في حال وجود أي قيد.

```
SQL > DROP CLUSTER Clu1
      INCLUDING TABLES;
```



إدارة التجمّعات باستخدام الأداة Schema Manager

كما في السابق، يمكن بسهولة إنشـــاء تجمّع جديد بالنقر بزر الفـــأرة الأيمـــن عـــلى العقدة Cluster في نافذة Schema Manager يظهـــر صـــندوق حوار Create Cluster، كما في الشكل ٣٢-٣.

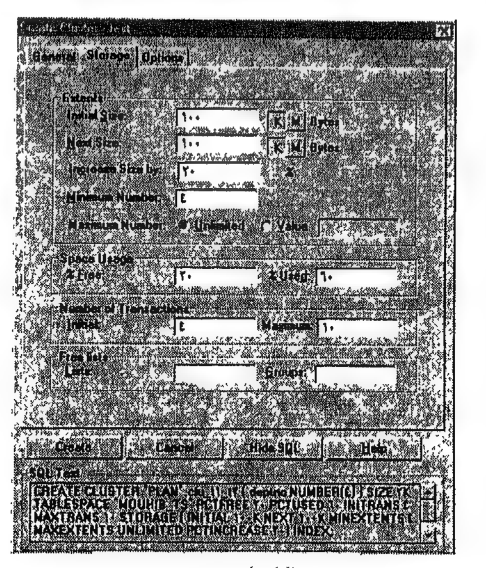
حـــدّد اســـم الـــتجمّع Name والمخطـــط الذي سيتم إنشاؤه فيه Schema وحجمه Size ونمـــط الـــتجمّع: Index Cluster أو Hash cluster حدّد أيضا أعمدة مفتاح التجمّع Cluster Key Columns.



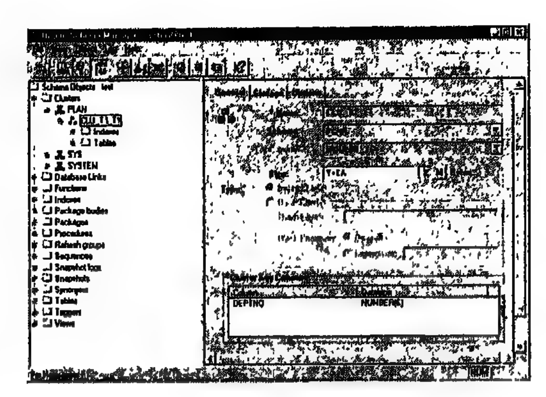
الشكل ٣٢-٣

تســـتطيع رؤيـــة تعـــليمة SQL الموافقة لتعـــليمة إنشـــاء الـــتجمّع بالنقر على زر Show SQL.

تســـتطيع بالانـــتقال إلى جـــزء الـــتبويب Storage تحديد وسطاء التخزين والمساحة المتعلّقة بهذا التجمّع (انظر الشكل ٣٢-٤).



الشكل ٣٢-٤



الشكل ٣٢-٥

تســتطيع أيضــاً تعديل تجمّع بالنقر عليه ضمن عقدة Cluster وإجراء الــتعديلات المطــلوبة فــي الجزء الأيمن من النافذة (انظر الشكل ٣٢-٥). أخيــراً تســتطيع حذف تجمّع بالنقر عليه بزر الفأرة الأيمن وطلب الخيار Remove.



التجمّعات وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للتجمّعات هي:

- المشاهد المتعلقة بالتجمّعات: *USER_CLUSTERS, DBA_CLUSTERS*

- المشاهد المتعلقة بأعمدة التجمّعات: *ALL_TAB_COLUMNS, USER_TAB_COLUMNS, DBA_TAB_COLUMNS*.
- المشاهد المتعلقة بالأعمدة المفاتيح للتجمّعات: *USER_CLU_COLUMNS, DBA_CLU_COLUMNS*.



إدارة الوحدات البرمجية

Administrating Program Unites

تستطيع ضمن أوراكل استخدام الوحدات البرمجية التالية:

- الإجرائيات Procedures.
- الدالات Functions.
- الحزم البرمجية Packages.

حيث يتم ترميز هذه الوحدات بلغة PL / SQL.

الإجرائيات Procedures

الإجـــرائية هـــي مجموعـــة من تعليمات PL / SQL التي تكوّن برامجاً جزئيةً. يمكن ألا تحـــتوي الإجـــرائية على أي وسيط دخل أو خرج، كما يمكن أن تحتوي على عدة وسطاء دخل أو وسطاء خرج.

الشكل العام للإجرائية هو:

```
Procedure procedure_name [(parameter_declaration )] IS
 [local declarations]
BEGIN
  PL /SQL Statements
[EXCEPTION
        optional Exception Handler (S0)]
END [procedure_ name];
```

حيث يأخذ parameter_declaration الشكل:

```
 Parameter_ name [IN \ OUT\ IN OUT] datatype
```

حيث:

* **IN**: تحدّد أن الوسيط هو وسيط دخل.
* **OUT**: تحدّد أن الوسيط هو وسيط خرج.
* **IN OUT**: تحدّد أن الوسيط هو وسيط دخل وخرج.



```
Procedure get_customer_id (
            Last IN VARCHAR2,
            First IN VARCHAR2,
            Cust_ id OUT number)
Begin
        SELECT id INTO cust_id
        FROM customer
        WHERE last_name = last
            AND first_name = first;
End get_customer_id;
```

الدالات Functions

الدالة كالإجرائية، مجموعة من تعليمات PL/SQL التي تكوّن برنامجاً جزئياً، لكنها تختلف عن الإجرائية في أنها ترجع قيمة خرج وحيدة فقط.

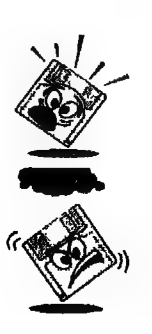
الشكل العام للدالة هو:

```
FUNCTION function_name [(parameter_type)]
        RETURN data type IS
[local declarations]
BEGIN
 PL /SQL statements
[EXCEPTION
   Optional Exception Handel (s) ]
END [ function_name];
```



```
FUNCTION get_customer_id (
    Last IN VARACHAR2,
    First IN VARCHAR2 )
RETURN INTEGER IS
        Eust_id INTEGER ;
BEGIN
 SELECT id INTO cust_id
FROM customers
WHERE last_name = last
AND first_name = first;
RETURN cust_id;
EXCEPTION
      WHEN OTHERS THEN
            RETURN NULL;
END get_customer_id;
```

يمكن باستخدام تعليمة ***CREATE OR REPLACE function or procedure*** إنشاء دالة أو إجرائية مباشرةً من خلال سطر الأوامر.



```
CREATE OR REPLACE PROCEDURE EMPLOYEES
AS
CURSOR emp_ cursor IS
        SELECT ename, sal, empno
        FROM emp
```

```
        WHERE sal > 1000;
BEGIN
        RDBMS_OUTPUT.PUT_LINE ('Employees has more than
1000sal' );
        RDBMS_OUTPUT. PUT_ LINE ('name, sal, empno )
        FOR emp IN emp_Cursor LOOP
                RDBMS_OUTPUT. PUT_LINE
                        (name\\ ' '\\ sal \\' '\\empno);
        END LOOP;
END Employees;
```

تستطيع الآن تنفيذ الإجرائية التالية على الشكل التالي:

```
SQL > set server output on,
SQL > execute employees;
```

الحزم البرمجية Packages

عبارة عن مجموعة من الإجرائيات والدالات المرتبطة والتي يتم ترجمتها وتخزينها سوية في قاموس المعطيات.

طبعاً تسمح لك الحزم البرمجية بتجميع أنماط PL/SQL والعناصر والبرامج الجزئية سويةً في وحدة منطقية.

وعندما تقوم بربط هذه العناصر، يمكن بسهولة برمجة وتعديل هذه الوحدات كما يتم تحسين الأداء لأنه يتم شحن كامل الحزمة البرمجية إلى الذاكرة عند أول طلب.

يتم إنشاء الحزم البرمجية من خلال جزأين:

١- **الجزء الأول:** هو توصيف الحزمة البرمجية ويأخذ الشكل:

```
CREATE PACKAGE package_name AS
                        Package_spicifications
Public type and object declaration
Subprogram definition
END [package_name]
```

٢- **الجزء الثاني:** وهو جزء كتابة محتوى الحزمة البرمجية ويأخذ الشكل:

```
CREATE PACKAGE BODY package_name AS
Package_body
Private type and object declaration
```

```
Subprogram bodies
[BEGIN
Initialization statements]
END [package_ name]

CREATE OR REPLACE PACKAGE part_ mgmt IS
        TYPE parts_ type IS REF CURSOR
        RETURN parts % ROW TYPE;
PROCEDURE insert_part_ (part_ record parts %
ROWTYPE);
PROCEDURE updat_part_ unitprice (part_id IN
INTEGER, new_price IN INTEGER)
FUNCTION get_part_ id (part_ desc IN VARCHAR 2)
        RETURN INTEGER;
END part_ mgmt;
CREATE OR REPLACE PACHKSAGE BODY part; mgmt IS
PROCEDURE insert_part (part_ record part % ROWTYPE )
        IS dup_primary_ key EXCEPTION;
PRAGMA EXCEPTION_ INIT (dup_primary_ key,_1);
BEGIN
        INSERT INTO parts
        VALUES (part_record. Id_part_ record. Unit_price, part_
record. Description );
EXCEPTION
        WHEN dup_primary_ key THEN raise_application_ error ( _
20001- 'Duplicate part ID');
        WHEN OTHERS THEN
        Raise_application_error (-20001, 'Undefind exception' );
END insert_part;

... other package procedure and function definitions....
END part_mgmt;
```

تستطيع مثلاً استخدام عناصر هذه الحزمة البرمجية على الشكل التالي:

```
SELECT INTO part_ mgmt. Current_part
FROM parts
WHERE id = 3 ;
Part_mgmgt. Insert_part (3,500, 00, 'Network Computer' );
```

الوحدات البرمجية وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للتجمّعات هي:

- ☆ المشاهد المتعلقة بأخطاء الترجمة: *USER_ERRORS, ALL_ERRORS, DBA_ERRORS*.
- ☆ المشاهد المتعلقة بأحجام العناصر: *DBA_OBJECT_SIZE, USER_OBJECT_SIZE*.
- ☆ المشاهد المتعلقة بنصوص مصادر العناصر: *DBA_SOURCE, ALL_SOURCE, USER_SOURCE*.




الباب التاسع



٣٤. أمان المعطيات.

٣٥. إدارة المستخدمين.

٣٦. إدارة الامتيازات.

٣٧. إدارة الوظائف.

٣٨. إدارة التشكيلات الجانبيّة.





أمان المعطيات
Data Security

من أهـم أهداف أمان قواعد المعطيات، اختيار السماح أو عدم السماح للمستخدمين بالوصـول إلى العناصر ضمن قاعدة المعطيات. أما إمكانية وصول المستخدم إلى العناصر فتتمّ من خلال الامتيازات الممنوحة له.

يقوم نظام أوراكل بإدارة الأمان في قواعد المعطيات من خلال عدة أدوات منها مخططات العناصر والمستخدمين.

فـلكل قـاعدة معطيـات قائمـة من مخططات العناصر وهي عبارة عن تجميع لعناصر المخططات مثل الجداول والمشاهد والتجمعات والإجرائيات والحزم البرمجية.

كذلك فإنّ لكل قاعدة معطيات قائمة من المستخدمين. وحتى يستطيع المستخدم الدخول إلى القـاعدة، يجـب عـليه تشغيل تطبيق قاعدة معطيات (مثل SQL*Plus) والاتصال بهذه القاعدة.

وعند إنشاء مستخدم قاعدة معطيات، ينشأ تلقائياً مخطط موافق بنفس الاسم خاصّ بهذا المستخدم. وبشكل افتراضي، عندما يقوم المستخدم بالاتصال مع القاعدة، يمكنه الوصول إلى جميع العناصر الموجودة في المخطط الموافق.

ولا يمكن للمستخدم الارتباط إلا مع مخطط من نفس الاسم، لذلك فإنّ مفهومي المستخدم والمخطط متطابقان.

ويتم التحكم بحقّوق المستخدم للدخول إلى قاعدة معطيات بتحديد القيم المختلفة المتعلقة بمجال أمان المستخدمين وتتضمن:

- فيما إذا كان استيقان Authentication معلومات المستخدم تحقّق من خلال قاعدة المعطيات أو من خلال نظام التشغيل.
- تحديد الفضاء الجدولي الافتراضي والمؤقت الخاص بالمستخدم.
- قائمة بالفضاءات الجدولية التي يمكن للمستخدم الدخول إليها ونصيب المستخدم في كل فضاء من هذه الفضاءات.
- القيود المتعلقة بمصادر المستخدم Profile.
- الامتيازات Privileges والوظائف الممنوحة للمستخدم.

استيقان المستخدم User Authentication

من أجل الابتعاد عن العمليات الممنوعة عن مستخدمي قاعدة معطيات. يقوم نظام أوراكل باستخدام طريقتين:

١- الاستيقان من خلال نظام التشغيل: في حال سمح نظام التشغيل بذلك، يمكن لنظام أوراكل استخدام المعلومات الموجودة في نظام التشغيل من أجل استيقان المستخدمين، هناك فوائد عديدة لذلك أهمها:

* يمكن للمستخدمين الاتصال مع قاعدة معطيات أوراكل بشكل أسهل بدون تحديد اسم المستخدم وكلمة المرور فمثلاً يمكن للمستخدم تشغيل SQL*plus بكتابة: /SQLPLUS
* يتركّز التحكّم بسماحيات المستخدم في نظام التشغيل، حيث لا يحتاج أوراكل إلى تخزين أو إدارة كلمات المرور المتعلقة بالمستخدمين.

ولتحديد استيقان مستخدم من خلال نظام التشغيل حدّد قيمة الوسيط OS_AUTHENT_PREFIX واستخدمه ضمن أسماء مستخدمي ORACLE، يحدد هذا الوسيط السابقة Prefix التي سيقوم نظام أوراكل بإضافتها إلى بداية اسم أي حساب لمستخدمي نظام التشغيل.

لنفترض مثلاً أننا حددنا قيمة هذا الوسيط على الشكل :

OS_AUTHENT_PREFIX = OPS$

فإذا كان لمستخدم حساب ضمن نظام التشغيل بالاسم "TSMOHIB" وأراد الاتصال مع قاعدة معطيات أوراكل وتحديد الاستيقان من قبل نظام التشغيل، سيقوم أوراكل بالتحقّق من وجود مستخدم قاعدة معطيات بالاسم "OPS$TSMOHIB" فإذا وجده سيسمح للمستخدم بإجراء عملية الاتصال.

٢ً-الاستيقان من خلال قاعدة معطيات أوراكل الموافقة، وذلك باستخدام المعلومات المخزنة في القاعدة.

تستخدم عادةً إحدى الطريقتين السابقتين، لكن يسمح نظام أوراكل باستخدام هاتين الطريقتين معاً ضمن نفس ممثل قاعدة المعطيات.

تحديد الفضاءات الجدولية للمستخدمين ونصيب كل مستخدم

يرتبط كل مستخدم بفضاء جدولي افتراضي، وعندما يقوم بإنشاء عنصر مخطط دون أن يحدد اسم الفضاء الجدولي، يتم تلقائياً استخدام الفضاء الجدولي الافتراضي.

كذلك يرتبط كل مستخدم بفضاء جدولي مؤقت ، يستخدم لتخزين المقاطع المؤقتة اللازمة أحياناً عند تنفيذ تعليمة SQL.

ويمكن تحديد نصيباً لكل مستخدم من كل فضاء جدولي مرتبط به وذلك لتحديد المساحة التي يستطيع أن يستخدمها من هذا الفضاء حيث يمكن استخدام مساحة تخزين محددة بالبايت (أوK الكيلو بايت أو الميغا بايت) أو استخدام مساحة غير محدّدة من هذا الفضاء.

ويمكن إلغاء وصول مستخدم إلى فضاء جدولي بتحديد نصيبه بالقيمة 0.

مجموعة المستخدم PUBLIC

تحتوي كل قاعدة معطيات على مجموعة مستخدم بالاسم public تسمح بالوصول العام إلى عناصر مخطط محددة (جداول، مشاهد،...إلخ) وتزوّد جميع المستخدمين بامتيازات نظام محددة.

وينتمي أيّ مستخدم تلقائياً إلى هذه المجموعة، حيث يمكنه رؤية جميع جداول قاموس المعطيات المتعلقة بـ ALL,USER. بالإضافة إلى ذلك يمكن للمستخدم منح امتياز أو وظيفة للمجموعة PUBLIC، ويستطيع جميع المستخدمين استخدام الامتيازات الممنوحة لهذه المجموعة.

توجد بعض القيود على مجموعة المستخدم PUBLIC أهمها:

- لا يمكن تحديد نصيب فضاء جدولي لهذه المجموعة، ولا يمكن إعطاؤها امتياز النظام UNLIMITED TABLESPACE.
- يمكن فقط إنشاء ارتباطات ومرادفات كعناصر عامة PUBLIC وذلك باستخدام CREAT PUBLIC DATABASE LINK/SYNONYM ولا يمكن إنشاء عناصر أخرى كعناصر عامة، فمثلاً التعليمة التالية غير صحيحة:

CREATE TABLE public.emp….;

مقيّدات مصادر المستخدم والتشكيلات الجانبية

مقيّدات المصادر *Resource Limits*

يمكن وضع قيود على مصادر النظام المختلفة الممنوحة للمستخدم وذلك لمنع أيّ تجاوز غير مقيد لمصادر النظام الهامة كزمن CPU مثلاً، هذه القيود مهمة جداً خاصة في الأنظمة متعددة المستخدمين، وتتمّ إدارة هذه القيود باستخدام تشكيلات المستخدم الجانبية user profile، وهي عبارة عن مجموعة مسمّاة لقيود المصادر التي يمكن وضعها لمستخدم.

أهم مقيدات مصادر النظام التي يمكن تحديدها:

- **مستوى الدورة Session level** : في أي وقت يقوم فيه المستخدم بالاتصال مع قاعدة المعطيات، يتم إنشاء دورة. وتحتاج كل دورة إلى جزء معين من زمن المعالج CPU ومن ذاكرة الحاسب الذي يعمل عليه النظام. وتوجد مجموعة من مقيدات مصادر النظام التي يمكن تحديدها على مستوى الدورة، فإذا تمّ تجاوز هذه المقيدات، يتم إنهاء التعليمة الحالية ويتم إرجاع رسالة توضح أنه قد تمّ الوصول إلى نهاية الدورة.

عند هذه النقطة، فإنّ جميع التعليمات السابقة في التحويل الحالي تبقى سليمة والتعليمات الوحيدة التي يمكن للمستخدم إنجازها هي ROLLBACK, COMMIT أو إلغاء الاتصال (حيث يتم في هذه الحالة تسجيل التحويل الحالي). أما العمليات الأخرى فإنها تعطينا رسالة خطأ.

حتى بعد أن يتم تثبيت commit أو إلغاء تثبيت roll back التحويل الحالي، لا يمكن للمستخدم إجراء أي عمل بشكل فعال خلال الدورة الحالية.

- **مستوى الطلب CALL level**: في أي وقت يتم فيه تنفيذ تعليمة SQL، يتم أخذ عدة خطوات لمعالجة هذه التعليمة. أثناء المعالجة هذه، يتم القيام بالعديد من الطلبات على قاعدة المعطيات كجزء من مراحل التنفيذ المختلفة. ومن أجل منع أي طلب من تجاوز حدود النظام بشكل كبير، فإنه يمكن تحديد مستوى الطلب. ففي حال تجاوز هذا المستوى، يتم إيقاف معالجة التعليمة ومن ثم إلغاء تسجيلها ويتمّ إرجاع رسالة خطأ، بينما تبقى بقيّة التعليمات السابقة في التحويل الحالي سليمة.
- **زمن المعالجة CPU time**: تنفيذ أية عملية ضمن أوراكل، تحتاج إلى زمن محدد من وقت CPU لمعالجة الطلب. ويمكن تحديد زمن CPU من أجل التحكم بالاستخدام غير المقيد لزمن CPU ويتم قياس مقيدات هذا الزمن بـ ٠.٠١ من الثانية.
- **عمليات القراءة المنطقية Logical Reads**: تعتبر عمليات الإدخال والإخراج I/O من أكثر العمليات صرفاً للزمن والذاكرة. ويمكننا وضع قيود على عمليات قراءة كتل المعطيات المنطقية خلال كل طلب وخلال كل دورة أيضاً حيث تتم القراءة من الذاكرة أو من القرص الصلب.

- الدورات المتزامنة Concurrent Sessions: يمكن تحديد عدد الدورات المتزامنة لكل مستخدم.
- زمن التوقف Idle Time: يمكن تحديد زمن التوقف لكل دورة. فإذا وصل الزمن بين طلبات أوراكل خلال الدورة إلى زمن التوقف، فسيتم إنهاء هذه الدورة وإلغاء تسجيل التحويل الحالي. وتتم أيضاً إعادة مصادر الدورة إلى النظام.
- زمن الاتصال Connect time: يمكن تحديد زمن الاتصال لكل دورة، فإذا تجاوزت الدورة هذا الزمن يتم إلغاؤها وإلغاء تسجيل التحويل الحالي، كذلك تتم إعادة مصادر الدورة إلى النظام.

التشكيل الجانبي Profile

هو مجموعة مسمّاة لمقيدات مصادر محدّدة يمكن ربطها مع مستخدم قاعدة معطيات أوراكل، لذلك فهي تعطينا أداة سهلة لإدارة مقيدات المصادر.

ونحتاج إلى إنشاء التشكيل الجانبي فقط عندما نكون بحاجة لحماية وإدارة قاعدة المعطيات.

من أجل استخدام التشكيل الجانبي، قم أولاً بتحديد المستخدمين المرتبطين ضمن قاعدة المعطيات ثم حدّد عدد التشكيلات الجانبية التي تحتاجها لجميع أنماط المستخدمين ضمن قاعدة المعطيات وأخيراً حدّد القيم المناسبة لمقيدات المصادر من أجل كل تشكيل جانبي.

الامتيازات Privileges

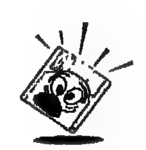
الامتياز هو حقّ لتنفيذ نمط خاص من تعليمات SQL أو للوصول إلى عنصر خاص بمستخدم آخر.

يمكننا إعطاء بعض الأمثلة عن الامتيازات:

- حقّ الاتصال بقاعدة المعطيات.
- حقّ إنشاء جدول.
- حقّ اختيار أسطر من جدول مستخدم آخر.
- حقّ تنفيذ إجرائية متعلقة بمستخدم آخر.

ويمكن للمستخدم الحصول على امتياز بطريقتين مختلفتين:

- الامتيازات التي يمكن منحها للمستخدمين بشكل خارجي explicitly، مثلاً الامتياز بإدراج سجلات إلى الجدول EMP يمكن منحه بشكل خارجي للمستخدم SCOTT.
- يمكن أن تمنح الامتيازات للوظائف (الوظيفة هي مجموعة مسمّاة من الامتيازات) ومن ثم يمكن منح الوظيفة لمستخدم أو أكثر. مثلاً امتيازات إدراج واختيار وتعديل وحذف سجلات من الجدول EMP يمكن أن تمنح لوظيفة بالاسم CLERK والتي هي أصلاً ممنوحة للمستخدمين SCOTT و BRIAN.



نظراً لأن الوظائف تسمح بإدارة الامتيازات بشكل أسهل وأفضل لذلك تمنح الامتيازات عادة للوظائف لا لمستخدمين محددين.

يوجد نمطان مختلفان من الامتيازات:

١- **امتيازات النظام System privileges**: وهي الامتيازات الخاصة بإجراء أعمال خاصة أو إجراء أعمال معينة على أنماط خاصة من العناصر.

وكمثال على امتيازات النظام، هناك امتيازات لإنشاء فضاءات جدولية، كذلك حذف أسطر من أي جدول من قاعدة المعطيات.

وهنالك أكثر من ٦٠ امتياز نظام يسمح كلٌ منها للمستخدم بإجراء عملية خاصة على قاعدة المعطيات أو مجموعة عمليات على القاعدة (انظر الملحق٣ لإلقاء نظرة على امتيازات النظام).

ويمكن منح grant أو إلغاء منح revoke امتيازات النظام من المستخدمين والوظائف. كذلك يمكن فقط للمستخدمين الذين يمتلكون امتياز النظام ADMIN OPTION أو المستخدمين الذين يمتلكون امتياز النظام GRANT ANY PRIVILEGE القيام بمنح أو إلغاء امتياز للمستخدمين والوظائف.

٢- **امتيازات العناصر Object privileges**: وهي الحقّ بإجراء عملية معينة على عناصر محددة كالجداول والمشاهد والسلاسل والإجرائيات والتوابع والحزم البرمجية.

فمثلاً امتياز حذف أسطر من الجدول DEPT هو امتياز عنصر. وتوجد أنواع مختلفة من امتيازات العناصر وفقاً لنمط كل عنصر.

وتمنح امتيازات العناصر عادةً للوظائف المعرّفة لمجموعات المستخدمين.

وبعض عناصر المخططات مثل التجمّعات والفهارس والقادحات وارتباطات قواعد المعطيات لا تمتلك امتيازات عناصر موافقة وإنما يتم التحكم بها باستخدام امتيازات النظام. فمثلاً لتعديل تجمّع يجب على المستخدم امتلاك هذا التجمّع أو امتلاك امتياز النظام ALTER ANY CLUSTER.

ويمتلك تلقائياً كل مستخدم جميع امتيازات العناصر المتعلقة به. ويمكنه منح أي امتياز عنصر لمستخدم آخر أو وظيفة أخرى.

وتسمح امتيازات العناصر الخاصة بالجداول بالتحكم بأمان الجداول ضمن مستويين:

١- العمليات المتعلقة بلغة التعامل مع المعطيات Data Manipulation Language Operations: تسمح امتيازات UPDATE, SELECT, INSERT, DELETE بإجراء عمليات DML على جدول. ويجب منح هذه الامتيازات فقط للمستخدمين أو للوظائف التي بحاجة للتعامل مع معطيات جدول.

ويمكن منح امتيازات الجداول UPDATE, INSERT لأعمدة محدّدة ضمن الجدول. حيث يمكن للمستخدم إدراج سطر فقط بالقيم المتعلقة بالأعمدة المحدّدة وتبقى بقية قيم الأعمدة الأخرى معدومة، نفس الأمر بالنسبة لتعديل قيم سطر.

٢- العمليات المتعلقة بلغة تعريف المعطيات Data Definition Language Operations: تسمح الامتيازات REFERNCES, INDEX, ALTER بإجراء عمليات DDL على جدول. ويمكن منح الامتياز REFERENCES لأعمدة محددة ضمن جدول.

كذلك فإن امتيازات العناصر المتعلقة بالمشاهد تسمح بإجراء العديد من عمليات DML. فمثلاً لإنشاء مشهد يجب أن تمتلك الامتياز CREATE VIEW أو CREATE ANY VIEW وهناك أيضاً العديد من امتيازات العناصر المتعلقة بالإجرائيات كامتيازات:

CREATE PROCEDURE, CREATE ANY PROCEDURE
ALTER ANY PROCEDURE
EXECUTE ANY PROCEDURE

يوضح الجدول التالي مجموعة امتيازات العناصر:

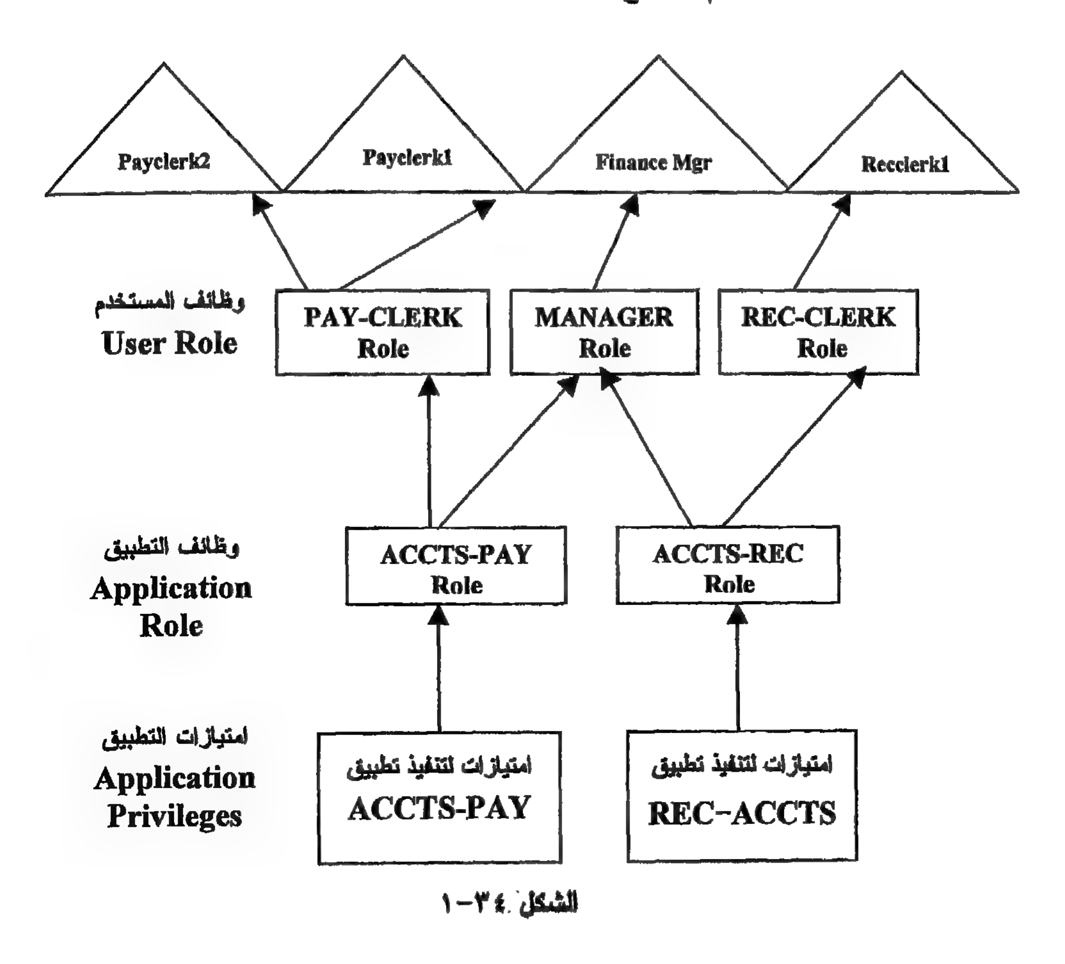
امتيازات العناصر	الجداول	المشاهد	المرادفات	الإجرائيات
ALTER	+		+	
DELETE	+	+		
EXECUTE				+
INDEX	+			
INSERT	+	+		
REFERENCESE	+			
SELECT	+	+	+	
UPDATE	+	+		

الوظائف Roles

الوظائف عبارة عن مجموعات مسمّاة من الامتيازات المرتبطة التي يمكن منحها للمستخدمين أو لوظائف أخرى.



ويتم إنشاء الوظائف لإدارة الامتيازات المتعلقة بتطبيق قاعدة معطيات أو بإدارة الامتيازات الخاصة بمجموعة مستخدم. يوضح الشكل ٣٤-١ استخدامات الوظائف:



الشكل ٣٤-١

تمتلك وظيفة التطبيق Application Role جميع الامتيازات اللازمة لتشغيل تطبيق قاعدة معطيات.

ويمكن للتطبيق امتلاك عدّة وظائف مختلفة، لكل منها مجموعة امتيازات معينة.

أما وظيفة المستخدم User Role فيتم إنشاؤها لمجموعة مستخدمي قاعدة معطيات حيث يمكن التحكم بها بمنحها وظائف تطبيق وامتيازات خاصة. ويمكن بعد ذلك منح هذه الوظيفة للمستخدمين المحددين.

توجد مجموعة من الخواص المرتبطة بالوظائف أهمها:

- يمكن منح امتيازات نظام أو امتيازات عنصر لوظيفة ما.
- يمكن منح وظيفة لوظائف أخرى، ولا يمكن منح وظيفة لنفسها ولا يمكن أيضاً أن تمنح بشكل دوّار Circular.
- يمكن منح وظيفة لأي مستخدم قاعدة معطيات.
- أية وظيفة تمنح لمستخدم تكون مؤهّلة enabled أو غير مؤهّلة disabled.

يمكن لأي مستخدم يمتلك امتياز النظام ANY ROLE GRANT منح أو إلغاء وظيفة، كذلك يمكن لمن يمتلك امتياز النظام ADMIN أن يقوم بمنح أو إلغاء الوظائف.

وضمن أي قاعدة معطيات، يجب أن يكون اسم الوظيفة وحيداً، ولا يمكن أن يتشابه أيضاً مع اسم مستخدم. والوظائف ليست جزءاً من المخططات، بالتالي فإن أي مستخدم قام بإنشاء وظيفة يمكن إلغاؤه دون أن يؤثر ذلك على الوظيفة.

توجد مجموعة من الوظائف المعرفة ضمن أوراكل والموضحة بالجدول التالي :

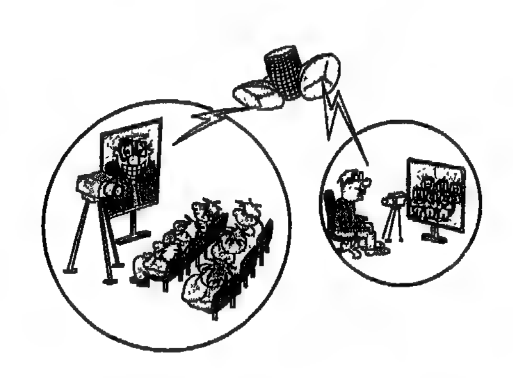
اسم الوظيفة	الامتيازات الممنوحة لها
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW.
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENSE, CREATE TABLE, CREATE TRIGGER
DBA	all system privileges WHITH ADMIN OPTION
EXP_FULL_DATABASE	SELECT ANY TABLE, BACKUP ANY TABLE, INSERT, DELETE, UPDATE on the table, SYS.INCVID, SYS.INCFIL and SYS.INCEXP
IMP_FULL_DATABASE	BECOME USER, WRITEDOWN

ويمكن للمستخدم تفعيل أي عدد من الوظائف التي يحتاجها باستخدام الوسيط MAX_ENABLED_ROLES.

توجد بعض الملاحظات المتعلقة بالارتباطات بين تعليمات لغة تعريف المعطيات DDL والوظائف أهمها:

- جميع امتيازات النظام وامتيازات العناصر التي تسمح لمستخدم بإجراء عمليات DDL يمكن أن تستخدم من خلال وظيفة. كمثال على ذلك امتيازات النظام: CREATE VIEW, CREATE TABLE, CREATE PROCEDURE وامتيازات العناصر INDEX, ALTER الخاصة بالجداول. أما الاستثناء فهو خاص بامتياز العناصر REFERENCES لجدول حيث لا يمكن استخدامه كتعريف لمفتاح ثان Foreign Key لجدول إذا استخدم من خلال وظيفة.
- جميع امتيازات النظام وامتيازات العناصر التي تسمح لمستخدم بإجراء عمليات DML المطلوبة لتوليد تعليمة DDL لا يمكن استخدامها من خلال وظيفة.

مثلاً إذا تلقى مستخدم امتياز النظام SELECT ANY TABLE أو امتياز العناصر SELECT لجدول وذلك من وظيفة، فلا يمكن استخدام أي منهما لإنشاء مشهد على جدول مستخدم آخر.







إدارة المستخدمين
Administrating Users

كما ذكرنا في الفصل السابق أنّه عند إنشاء مستخدم قاعدة معطيات، ينشأ تلقائياً مخطط موافق بنفس الاسم خاص بهذا المستخدم. وعندما يقوم المستخدم بالاتصال مع القاعدة، يمكنه الوصول إلى جميع العناصر الموجودة في المخطط الموافق.

كما يرتبط كل مستخدم بفضاء جدولي افتراضي، وعندما يقوم بإنشاء عنصر مخطط دون أن يحدد اسم الفضاء الجدولي، يتم تلقائياً استخدام الفضاء الجدولي الافتراضي SYSTEM.

كذلك يرتبط كل مستخدم بفضاء جدولي مؤقت، يستخدم لتخزين المقاطع المؤقتة اللازمة أحياناً عند تنفيذ تعليمة SQL.

ويمكن تحديد نصيباً لكل مستخدم من كل فضاء جدولي مرتبط به وذلك لتحديد المساحة التي يستطيع أن يستخدمها من هذا الفضاء حيث يمكن مساحة تخزين محدّدة بالبايت (أوK الكيلو بايت أو M الميغا بايت) أو اختيار مساحة غير محددة من هذا الفضاء.

ويمكن إلغاء وصول مستخدم إلى فضاء جدولي بتحديد نصيبه بالقيمة 0.

إنشاء مستخدمين جدد Creating New Users

يمكن القيام بذلك باستخدام الأداة Enterprise Manager أو Security Manager أو Server Manager. طريقة إدارة المستخدمين باستخدام Security Manger تشبه كثيراً طريقة الإدارة باستخدام الأداة Enterprise Manager إلا أنها تختلف عنها في أنها تقوم بذلك في قاعدة المعطيات الحالية فقط.

إنشاء مستخدم جديد باستخدام الأداة Server Manager

يمكنك إنشاء مستخدم جديد باستخدام تعليمة Create User على الشكل:

```
CREATE USER user_name
IDENTIFIED BY password |
IDENTIFIED EXTERNALY |
IDENTIFIED GLOBALY AS 'CN=user'
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA [number [K|M] | UNLIMITED] ON tablespace]
[, QUOTA [number [K|M] | UNLIMITED] ON tablespace]
[PROFILE profile]
[PASSWORD EXPIRE]
[ACCOUNT LOCK | ACCOUNT UNLOCK]
```

حيث:

- ☆ **user_name**: اسم المستخدم المطلوب إنشاؤه.
- ☆ **IDENTIFIED BY password**: لتحديد كلمة مرور للمستخدم.

☆ **IDENTIFIED EXTERNALLY**: لتحديد استيقان المستخدم من قبل نظام التشغيل. ويجب أن يكون اسم المستخدم مطابقاً لاسم المستخدم المعرّف في نظام التشغيل OS.

☆ **IDENTIFIED GLOBALLY AS 'CN=user'**: لتحديد استيقان المستخدم من قبل مجال أمان أوراكل. وتحدّد العبارة CN=user اسم المستخدم الخارجي.

☆ **DEFAULT TABLESPACE tablespace**: اسم الفضاء الجدولي الذي سيستخدمه المستخدم بشكل افتراضي. وفي حال لم يتم تحديده سيستخدم الفضاء الجدولي SYSTEM.

☆ **TEMPORARY TABLESPACE tablespace**: اسم الفضاء الجدولي المؤقت الذي سيستخدمه المستخدم بشكل افتراضي. وفي حال لم يتم تحديده سيستخدم الفضاء الجدولي SYSTEM.

☆ **QUOTA**: لتحديد نصيب المستخدم في الفضاء الجدولي إما بقيمة محددةK أو M، أو بقيمة غير محددة UNLIMITED.

☆ **PROFILE**: لتحديد اسم التشكيل الجانبي الخاص بالمستخدم.

☆ **PASSWORD EXPIRE**: لجعل كلمة المرور غير فعّالة مباشرة بعد إنشاء المستخدم، ويجب عليه في هذه الحالة تغيير كلمة المرور قبل أن يستطيع الدخول إلى القاعدة.

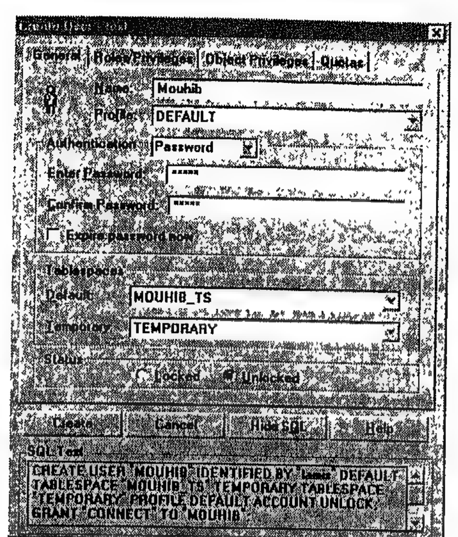
☆ **ACCOUNT LOCK**: لقفل حساب المستخدم بعد إنشائه.

☆ **ACCOUNT UNLOCK**: لإلغاء قفل حساب المستخدم بعد إنشائه.



```
CREATE USER mohib
IDENTIFIED BY yazan
DEFAULT TABLESPACE mohib_ts
TEMPORARY TABLESPACE tmp_ts
QUOTA 10 M ON mohib_ts
PROFILE default;
```

إنشاء مستخدم جديد باستخدام الأداة Security Manager



الشكل ٣٥-١

عند تشغيل برنامج Security Manager، قم بتوسيع عقدة Users تظهر مجموعة أسماء المستخدمين الموجودين في القاعدة إضافة إلى المعلومات المتعلقة بكل منهم.

انقر بزر الفأرة الأيمن على هذه العقدة ثم اختر الأمر Create User يظهر صندوق حوار مشابه للشكل ٣٥-١.

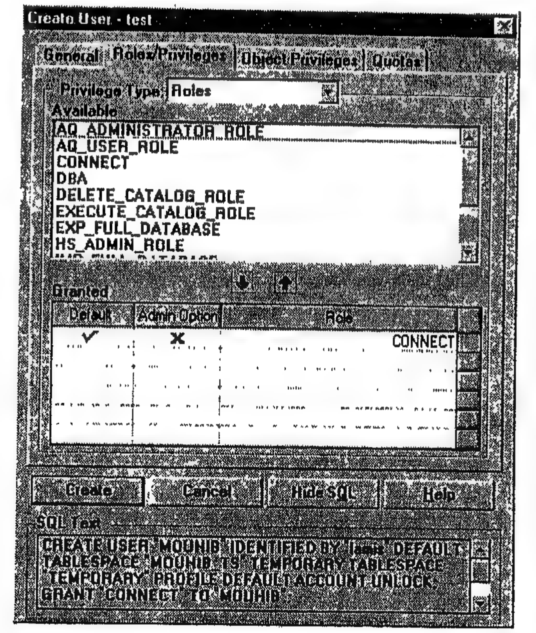
في جزء General حدّد اسم المستخدم والتشكيل الجانبي الموافق Profile، كذلك كلمة المرور Password.

حدّد أيضاً الفضاء الجدولي الافتراضي Default Tablespace المرتبط بالمستخدم، والفضاء الجدولي المؤقت الخاص به Temporary Tablespace. حدّد كذلك حالة المستخدم Locked أو Unlocked.

قم بتعبئة القيم المطلوبة ثم انقر زر OK لإنشاء المستخدم.

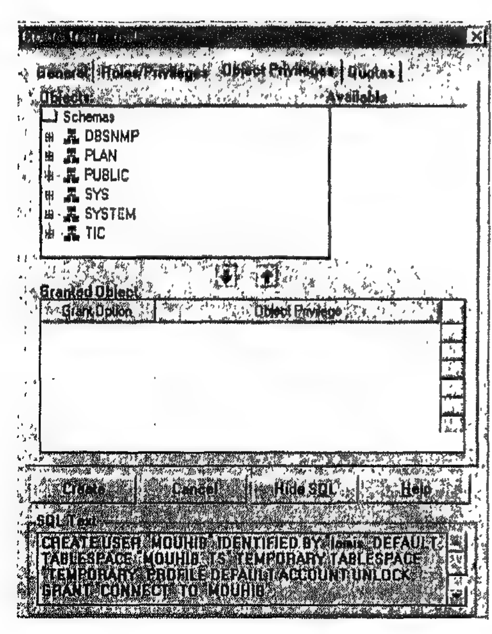
أما في الجزء الثاني Roles/Privileges فيمكنك تحديد الامتيازات والوظائف الممنوحة للمستخدم (انظر الشكل ٣٥-٢).

(انظر الفصل ٣٦ والفصل ٣٧ لمزيد من التفاصيل حول الامتيازات والوظائف).



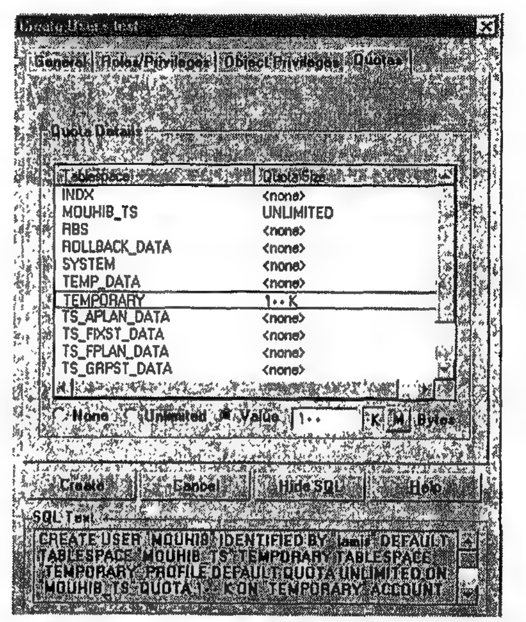
الشكل ٣٥-٢

في الجزء الثالث Object Privileges، تستطيع تحديد امتيازات العناصر التي يمكن منحها للمستخدم (انظر الشكل ٣٥-٣).
انظر الفصل ٣٦ لمزيد من التفاصيل حول امتيازات العناصر.



الشكل ٣٥-٣

أخيراً يمكنك في الجزء Quotas تحديد نصيب المستخدم في الفضاء الجدولي الافتراضي والفضاء الجدولي المؤقت (انظر الشكل ٣٥-٤).



الشكل ٣٥-٤

تعديل المستخدمين Altering Users

يمكن القيام بذلك باســتخدام الأداة Enterprise Manager أو Security Manager أو Server Manager.

تعديل مستخدم باستخدام الأداة Server Manager

يمكنك تعديل مستخدم باستخدام تعليمة ALTER User على الشكل:

```
ALTER USER user_name
IDENTIFIED BY password |
IDENTIFIED EXTERNALY |
IDENTIFIED GLOBALY AS 'CN=user'
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
[QUOTA [number [K|M] | UNLIMITED] ON tablespace]
[, QUOTA [number [K|M] | UNLIMITED] ON tablespace]
[PROFILE profile]
[PASSWORD EXPIRE]
[ACCOUNT LOCK | ACCOUNT UNLOCK]
[DEFAULT ROLE role[,role] |
[DEFAULT ROLE ALL[EXPIRE role[,role]]] |
[DEFAULT ROLE NONE]
```

حيث:

☆ **user_name**: اسم المستخدم المراد تعديله.

☆ **IDENTIFIED BY password**: لتحديد كلمة مرور المستخدم.

☆ **IDENTIFIED EXTERNALLY**: لتحديد استيقان المستخدم من قبل نظـام التشغيل. ويجب أن يكون اسم المستخدم مطابقاً لاسم المستخدم المعــرّف فــي نظــام التشغيل OS.

☆ **IDENTIFIED GLOBALLY AS 'CN=user'**: لتحديــد اســـتيقان المستخدم من قبل مجال أمان أوراكل. وتحدد العبـــارة CN=user اســم المســتخدم الخارجي.

☆ **IDENTIFIED GLOBALLY AS 'CN=user'**: لتحديد استيقان المستخدم من قبل مجال أمان أوراكل. وتحدد العبارة CN=user اسم المستخدم الخارجي.

☆ **DEFAULT TABLESPACE tablespace**: اسم الفضاء الجدولي الذي سيستخدمه المستخدم بشكل افتراضي. وفي حال عدم تحديده سيستخدم الفضاء الجدولي SYSTEM.

☆ **TEMPORARY TABLESPACE tablespace**: اسم الفضاء الجدولي المؤقت الذي سيستخدمه المستخدم بشكل افتراضي. وفي حال عدم تحديده سيستخدم الفضاء الجدولي SYSTEM.

☆ **QUOTA**: لتحديد نصيب المستخدم في الفضاء الجدولي إما بقيمة محددة K أو M. أو بقيمة غير محددة UNLIMITED.

☆ **PROFILE**: لتحديد اسم التشكيل الجانبي الخاص بالمستخدم.

☆ **PASSWORD EXPIRE**: لجعل كلمة المرور غير فعالة مباشرة بعد إنشاء المستخدم، ويجب عليه في هذه الحالة تغيير كلمة المرور قبل أن يتمكّن من الدخول إلى القاعدة.

☆ **ACCOUNT LOCK**: لقفل حساب المستخدم بعد إنشائه.

☆ **ACCOUNT UNLOCK**: لإلغاء قفل حساب المستخدم بعد إنشائه.

☆ **DEFAULT ROLE role [,role]**: لتحديد الوظيفة الافتراضية أو وظائف لهذا المستخدم.

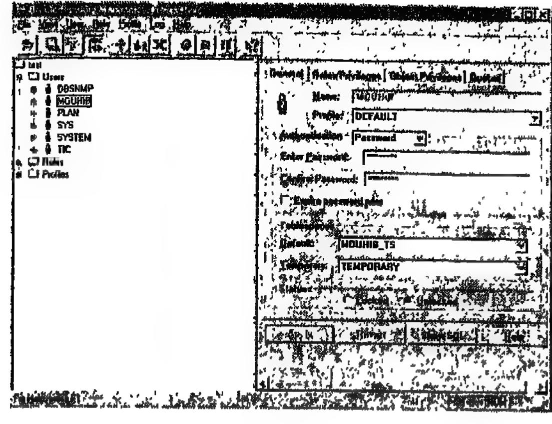
☆ **DEFAULT ROLE ALL**: لتأهيل جميع الوظائف للمستخدم، ويمكن باستخدام الخيار EXCEPT role[,role] إلغاء تأهيل بعض الوظائف.

☆ **DEFAULT ROLE NONE**: لعدم تأهيل أي وظيفة لهذا المستخدم.



```
ALTER USER mohib
IDENTIFIED BY yazan
QUOTA 20M ON mohib_ts
```

تعديل مستخدم باستخدام الأداة Security Manager



الشكل ٣٥-٥

عــند تشغيل برنامج Security Manager، قــم بتوســيع عقدة Users تظهــر مجموعة أسماء المســتخدمين الموجوديــن فــي القــاعدة إضــافة إلى المعلومات المتعلقة بكل منهم كما في الشكل ٣٥-٥.

انقر على المستخدم المراد تعديله، تظهــر فــي الــناحية اليمــنى المعــلومات المتعــلقة بهــذا المستخدم. تستطيع هنا القيام بالتعديلات المطلوبة عدا الاسم.

حذف المستخدمين Dropping Users

حذف مستخدم باستخدام الأداة Server Manager

يمكنك تعديل مستخدم باستخدام تعليمة ALTER User على الشكل:

DROP USER user_name [CASCADE]

حيث:

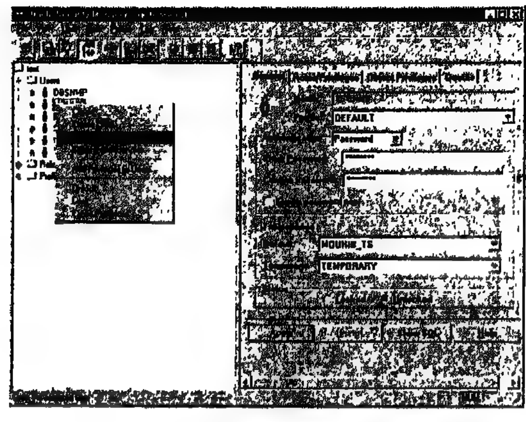
- ☆ **user_name**: اسم المستخدم المراد حذفه.
- ☆ **CASCADE**: لحذف جميع العناصر الموجودة ضمن مخطط المستخدم قبل حــذف المســتخدم نفســه. ويجب تحديد هذه الخيار في حال لم يكن مخطط عناصر المستخدم فارغاً.



DROP USER mohib CASCADE;

حذف مستخدم باستخدام الأداة Security Manager

انقر بزر الفـأرة الأيمـن علـى أيقونة المستخدم المراد حذفه، ثـم اختر الأمر REMOVE كما في الشكل ٣٥-٦.



الشكل ٣٥-٦

إدارة الدورات Administrating Sessions

كما نعلم فإن الدورة عبارة عن مجموعة من الأحداث التي تحصل منذ لحظة قيام المستخدم بالاتصال مع أوراكل وحتى قيامه بإلغاء هذا الاتصال.

لكل نسخة من نسخ أوراكل عدد أعظمي من دورات العمل (يتم تحديده عند الشراء) ويمكن تحديده من خلال الوسيط :

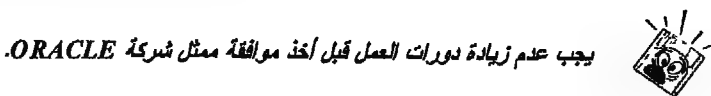
LICENSE_MAX_SESSIONS

ويوجد وسيط آخر LICENSE_SESSIONS_WARNING يسـاعد علـى إعطـاء رسائل تحذير عند وصول عدد الدورات في القاعدة إلى العدد الأعظم.

يمكن استخدام تعليمة ALTER SYSTEM لتغيير قيم هذين الوسيطين مثلاً:



```
ALTER SYSTEM
SET LICENSE_MAX_SESSIONS = 64
LICENSE_SESSIONS_WARNING = 54;
```


يجب عدم زيادة دورات العمل قبل أخذ موافقة ممثل شركة ORACLE.

يوجـد أيضاً وسيط آخر LICENSE_MAX_USERS يحدد عدد المستخدمين الأعظمي للقاعدة ويمكن تغييره أيضاً بنفس الطريقة السابقة، مثلاً :



ALTER SYSTEM
SET LICENSE_MAX_USERS = 300;

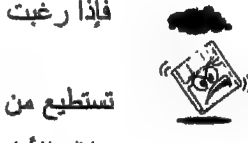


طبعاً من غير المسموح به تغيير و زيادة قيمة هذا الوسيط قبل أخذ موافقة شركة ORACLE.

وفي حال حدوث أي مشكلة في أي دورة من الدورات، فيمكن حذف هذه الدورة باستخدام تعليمة ... ALTER SYSTEM KILL SESSION، مثلاً يمكننا معرفة المعلومات المتعلقة بالدورات الفعالة على الشكل:



SELECT sid, serial#, username
FROM v$session;

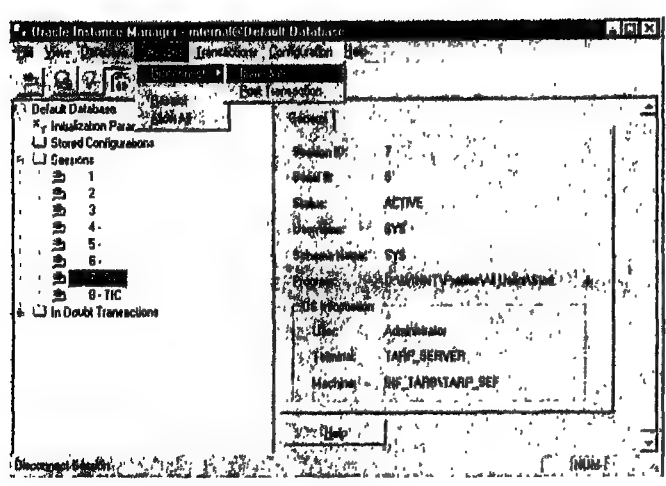


فإذا رغبت بحذف الدورة التي محددها sid=٨ و serial#=٢٣ نفّذ التعليمة:
ALTER SYSTEM KILL SESSION '8,23';

تستطيع من خلال الأداة Oracle Instance Manager معرفة الدورات، وخصائص كل دورة من هذه الدورات (انظر الشكل ٣٥-٧).

الشكل ٣٥-٧

يمكنك بسهولة إلغاء اتصال أي مستخدم بطلب الأمر Disconnect من قائمة Sessions، كما في الشكل ٣٥-٨.

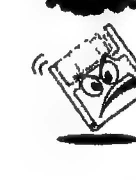


الشكل ٣٥-٨

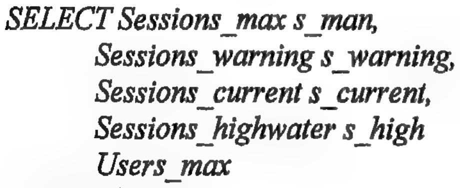
المستخدمون وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للمستخدمين هي:

- ☆ المشاهد المتعلقة بالمستخدمين: USER_USERS, ALL_USERS, DBA_USERS.
- ☆ المشاهد المتعلقة بالاتصال والترخيص والدورات: V$CIRCUIT, V$LICENCE, V$SESSION, V$SESSION_WAIT.



يمكن بإجراء استفسار على مشهد قاموس المعطيات V$LICENSE معرفة قيم الوسطاء السابقة :



```
SELECT Sessions_max s_man,
       Sessions_warning s_warning,
       Sessions_current s_current,
       Sessions_highwater s_high
       Users_max
FROM v$license;
```



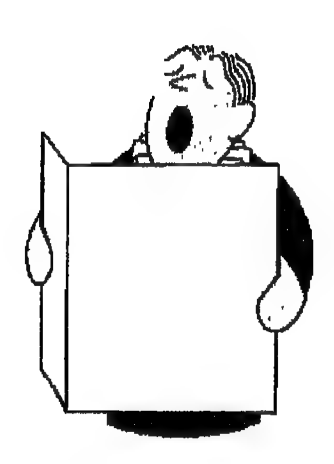
إظهار قائمة جميع المستخدمين مع معلومات عنهم :

*SELECT * FROM sys.dba_users;*



إظهار نصيب المستخدم في الفضاء الجدولي:

SELECT * FROM sys.dba_ts_quotas;





إدارة الامتيازات
Administrating Privileges

الامتياز هــو حق لتنفيذ نمط خاص من تعليمات SQL أو للوصول إلى عنصر خــاص بمســتخدم آخــر. كحق الاتصال بقاعدة المعطيات وحق إنشاء جــدول وحــق اختيار أسطر من جدول مستخدم آخر وغيرها. ويمكن للمستخدم الحصول على امتياز بطريقتين مختلفتين:

- ❑ الامتيازات التي يمكن منحها للمستخدمين بشكل خارجي.
- ❑ الامتيازات التي يمكن أن تمنح للوظائف.

ويوجد نمطان مختلفان من الامتيازات: امتيازات النظام System privileges وامتيازات العناصر Object privileges.

كمــا يمتلك كل مستخدم تلقائياً جميع امتيازات العناصر المتعلقة به. ويمكنه منح أي امتياز عنصر يمتلكه لمستخدم آخر أو وظيفة أخرى.

منح امتيازات النظام Granting System Privileges

يمكـــن القيام بذلك باستخدام الأداة Enterprise Manager أو Security Manager أو Server Manager. طـــريقة إدارة الامتيازات باستخدام Security Manger تشبه كثيراً الطـــريقة باستخدام الأداة Enterprise Manager إلا أنها تختلف عنها في أنها تقوم بذلك في قاعدة المعطيات الحالية فقط.

منح امتياز نظام باستخدام الأداة Server Manager

يمكنك منح امتياز نظام لمستخدم أو وظيفة باستخدام تعليمة Grant على الشكل:

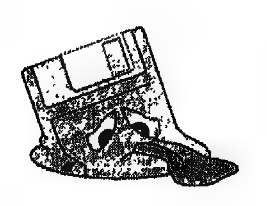
GRANT [system_privileges|role[,system_privileges|role]] TO
[user|role|PUBLIC[,user|role|PUBLIC]
[WITH ADMIN OPTION];

حيث:

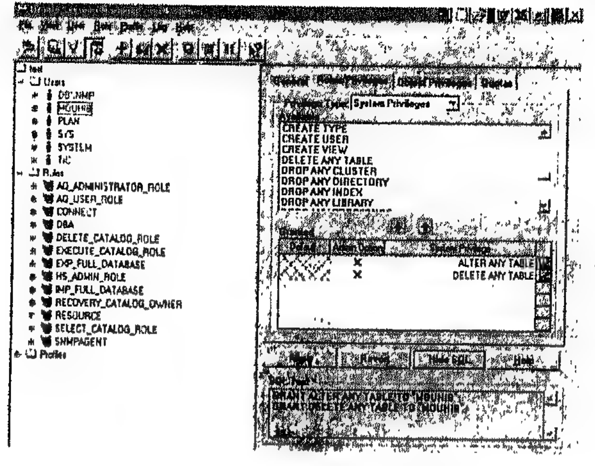
- ☆ **system_priviliges|role**: امتيازات النظام أو الوظائف المطلوب منحها.
- ☆ **user**: اسم المستخدم الذي سيتم منح الامتيازات له.
- ☆ **role**: اسم الوظيفة التي سيتم منح الامتيازات لها.
- ☆ **PUBLIC**: لمنح الامتيازات لكل المستخدمين.
- ☆ **WITH ADMIN OPTION**: للســماح للمســتخدمين أو الوظــائف بامتلاك سماحية منح أو إلغاء الامتيازات أو الوظائف.



GRANT ALTER USER,DROP USER TO mohib;

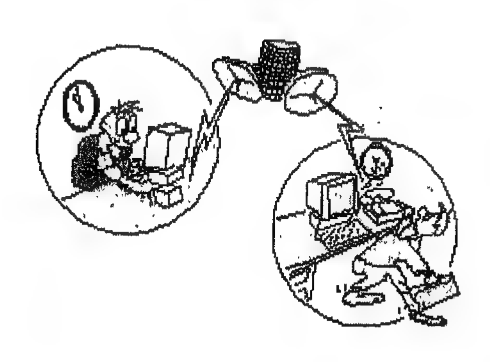


منح امتياز نظام باستخدام الأداة Security Manager



الشكل ٣٦-١

عندما ترغب بمنح امتياز نظام لمستخدم أو وظيفة، حدّد في العقدة Users المستخدم المطلوب منحه الامتياز، تظهر في الجزء الأيمن خصائص هذا المستخدم. انقر زر التبويب Roles/Privileges تظهر نافذة تشبه الشكل ٣٦-١. في صندوق التحرير والسرد Privilege Type حدّد القيمة System Privileges، حدّد بعدها الامتيازات التي ترغب بمنحها للمستخدم وانقر زر Apply.



إلغاء امتيازات النظام Revoking System Privileges

إلغاء امتياز نظام باستخدام الأداة Server Manager

يمكنك إلغاء امتياز نظام باستخدام تعليمة REVOKE على الشكل:

```
REVOKE [system_privileges|role[,system_privileges|role]] TO
        [user|role|PUBLIC[,user|role|PUBLIC]
[WITH ADMIN OPTION];
```

حيث:

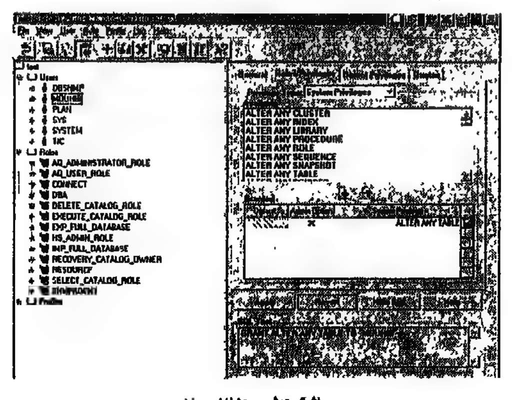
- ☆ **system_priviliges|role**: امتيازات النظام أو الوظائف المطلوب إلغاءها.
- ☆ **user**: اسم المستخدم الذي سيتم إلغاء الامتيازات منه.
- ☆ **role**: اسم الوظيفة التي سيتم إلغاء الامتيازات منها.
- ☆ **PUBLIC**: لإلغاء الامتيازات من كل المستخدمين.



REVOKE ALTER USER,DROP USER TO mohib;

إلغاء امتياز نظام باستخدام الأداة Security Manager

تستطيع ببساطة إلغاء امتياز نظام باستخدام الأداة Security Manager، حدّد المستخدم أو الوظيفة التي ترغب بإلغاء امتياز نظام منها، تظهر نافذة تشبه الشكل ٣٦-٢. انتقل إلى صفحة التبويب Roles/Privileges ثمّ قم بإلغاء الامتياز المطلوب.



الشكل ٣٦-٢

منح امتيازات عنصر Granting Object Privileges

منح امتيازات عنصر باستخدام الأداة Server Manager

يمكنك منح امتيازات عنصر لمستخدم باستخدام تعليمة Grant على الشكل:

```
GRANT object_privileges | ALL [(column[,column])]
        [,[object_privileges | ALL] [(column[,column])]]
        ON [schema.] object TO
        [user|role|PUBLIC[,user|role|PUBLIC]
[WITH ADMIN OPTION];
```

حيث:

- ☆ **object_priviliges**: امتيازات العناصر المطلوب منحها.
- ☆ **ALL**: لاختيار جميع امتيازات العنصر.
- ☆ **column**: لتحديد العمود الذي ستمنحه الامتيازات، وذلك مع الامتيازات INSERT، REFERENCES، UPDATE.
- ☆ **ON**: لتحديد العنصر الذي ستمنحه الامتيازات.
- ☆ **TO**: لتحديد أسماء المستخدمين الذين ستمنحهم امتيازات العناصر.
- ☆ **user**: اسم المستخدم الذي ستمنحه الامتيازات.
- ☆ **PUBLIC**: لمنح الامتيازات لكل المستخدمين.
- ☆ **WITH GRANT OPTION**: للسماح للمستخدمين بامتلاك سماحية منح أو إلغاء الامتيازات أو الوظائف.

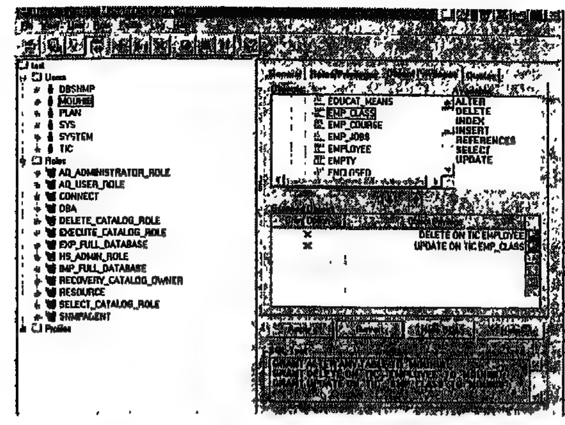


```
GRANT SELECT ON employee TO mohib, lamis;

GRANT SELECT,INSERT(emp_id,name), UPDATE(name)
ON employee
TO mohib;
```

منح امتيازات عنصر باستخدام الأداة Security Manager

عـــندما تـــرغب بمـــنح امتياز نظام لمستخدم أو وظيفة، حدّد في العقدة Users المستخدم المطـــلوب منحه الامتياز، تظهر في الجزء الأيمن خصائص هذا المســـتخدم. انقـــر زر التبويب Object Privileges تظهـــر نـــافذة تشـــبه الشـــكل ٣٦-٣. تستطيع هنا منح امتياز العنصر.



الشكل ٣٦-٣

إلغـــاء امتيازات عنصر

Revoking Object Privileges

إلغاء امتيازات عنصر باستخدام الأداة Server Manager

يمكنك إلغاء امتيازات عنصر باستخدام تعليمة REVOKE على الشكل:

```
REVOKE [object_privileges[,object_privileges]] ON [schema.]object
       FROM [user|role|PUBLIC[,user|role|PUBLIC]
       [CASCADE CONSTRAINTS];
```

حيث:

- ☆ **object_priviliges**: امتيازات العنصر المطلوب إلغاءها.
- ☆ **ON**: لتحديد العنصر الذي ستحذف امتيازات العنصر منه.
- ☆ **FROM**: أسماء المستخدمين أو الوظائف التي ستلغى الامتيازات منها.
- ☆ **PUBLIC**: لإلغاء الامتيازات من كل المستخدمين.

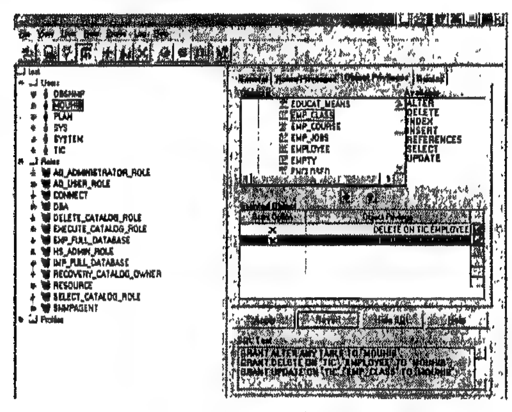
☆ CASCADE CONSTRAINTS: لإلغــاء قيــود التكامل المرجعي المعرّفة باستخدام الامتياز REFERENCES الذي سيتم حذفه.



REVOKE SELECT ON employee FROM mohib, lamis;

إلغاء امتيازات عنصر باستخدام الأداة Security Manager

تســتطيع ببساطة إلغاء امتياز نظام باســـــتخدام الأداة Security Manager، حــدّد المســتخدم أو الوظيفــة التي ترغب بإلغاء امتياز العنصــر مــنها، تظهر نافذة تشبه الشــكل ٣٦-٤. انتقل إلى صفحة التــبويب Object Privileges ثمّ قم بإلغاء امتياز العنصر المطلوب.



الشكل ٣٦-٤

الامتيازات وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للامتيازات هي:

☆ المشاهد المتعلقة بالامتيازات المحدّدة من قبل دورة: SESSION_PRIVS.

☆ المشــــاهد المتعــــلقة بامـــــتيازات الـــــنظام: DBA_SYS_PRIVS, USER_SYS_PRIVS.

☆ المشـــاهد المتعـــلقة بامـــتيازات الأعمــــدة: DBA_COL_PRIVS, ALL_COL_PRIVS, USER_COL_PRIVS, COLUMN_PRIVILIGES, USER_COL_PRIVS_MADE,

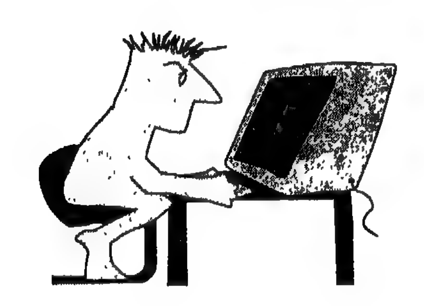
ALL_COL_PRIVS_MADE, USER_COL_PRIVS_RECD, ALL_COL_PRIVS_RECD.

☆ المشاهد المتعلقة بامتيازات العناصر: DBA_TAB_PRIVS, USER_TAB_PRIVS, ALL_TAB_PRIVS, USER_TAB_PRIVS_MADE, ALL_TAB_PRIVS_MADE, USER_TAB_PRIVS_RECD, ALL_TAB_PRIVS_RECD, TABLE_PRIVILEGES.

☆ المشاهد المتعلقة بامتيازات النظام المسموحة: V$ENABLEDPRIVS.



```
SELECT * FROM DBA_SYS_PRIVS
ORDER BY grantee, privilege;
```







إدارة الوظائف
Administrating Roles

الوظيفة عبارة عن مجموعة مسمّاة من الامتيازات المرتبطة التي يمكن منحها للمستخدمين أو لوظائف أخرى.

ويتم إنشاء الوظائف لإدارة الامتيازات المتعلّقة بتطبيق قاعدة معطيات أو بإدارة الامتيازات الخاصة بمجموعة مستخدم.

إنشاء الوظائف Creating Roles

يمكـن القيام بذلك باستخدام الأداة Enterprise Manager أو Security Manager أو Server Manager. طـريقة إدارة الامتيازات باستخدام Security Manger تشبه كثيراً طـريقة الإدارة باستخدام الأداة Enterprise Manager إلا أنها تختلف عنها في أنها تقوم بذلك في قاعدة المعطيات الحالية فقط.

إنشاء وظيفة باستخدام الأداة Server Manager

يمكنك إنشاء وظيفة باستخدام تعليمة CREATE ROLE على الشكل:

```
CREATE ROLE role
[NOT IDENTIFIED]
[IDENTIFIED BY password | EXTERNALLY | GLOBALLY];
```

حيث:

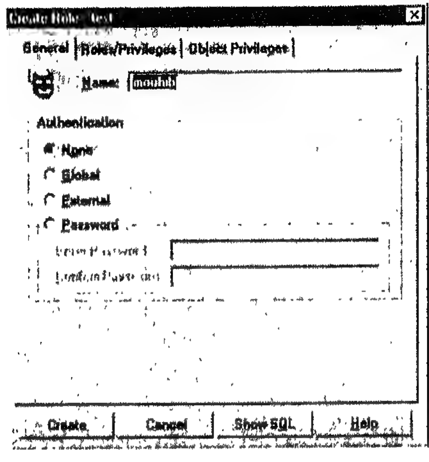
- ☆ **role**: اسم الوظيفة التي سيتم إنشاؤها.
- ☆ **NOT IDENTIFIED**: لـتحديد أن المستخدمين الذي منحوا الوظيفة لن يحتاجوا كي يتحقّق منهم مخدّم أوراكل لتأهيل تلك الوظيفة.
- ☆ **IDENTIFIED**: لـتحديد أن المستخدمين الذي منحوا الوظيفة سيحتاجون لكي يتحقق منهم مخدّم أوراكل لتأهيل الوظيفة.
- ☆ **BY password**: لتحديد كلمة المرور التي يجب إعطاؤها لتأهيل الوظيفة.
- ☆ **EXTERNALLY**: كي يقوم مخدّم أوراكل بالتحقق من وصول المستخدم للوظيفة باستخدام أدوات نظام التشغيل.



```
CREATE ROLE mohib;
```

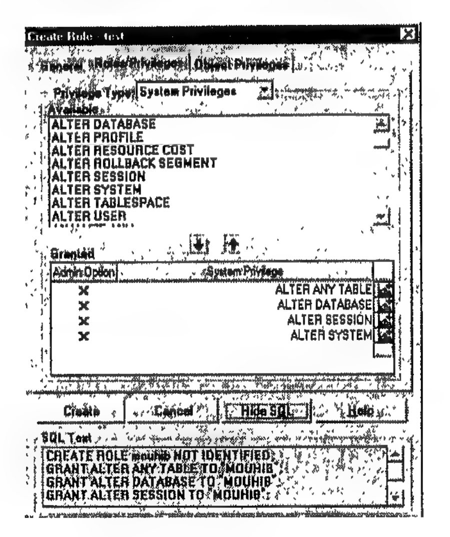
إنشاء وظيفة باستخدام الأداة Security Manager

انقر بزر الفأرة الأيمن على العقدة Roles ثـم اخـتر Create يظهر صندوق حوار مشـابه للشكل ٣٧-١.اكتب اسم الوظيفة وحدد الاستيقان الخاص بها.



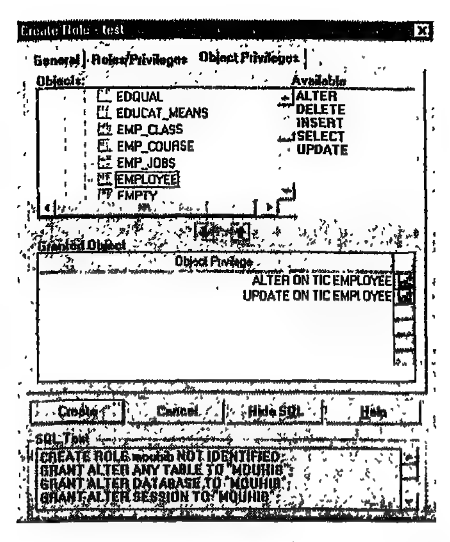
الشكل ٣٧-١

انقر على زر التبويب Roles/Privileges، يمكـنك مـن خلال صفحة التبويب هذه منح وظيفـة أو أكـثر لـلوظيفة التي تمّ إنشاؤها بسحب الوظائف المطلوبة وإفلاتها على عقدة الوظيفـة الجديدة. يمكنك بنفس الطريقة منح امتيازات النظام للوظيفة الجديدة. انظر الشكل ٣٧-٢.



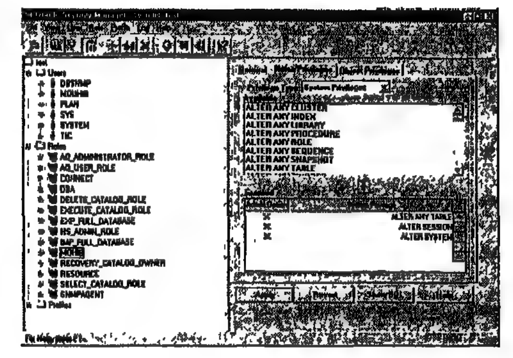
الشكل ٣٧-٢

انقر على زر التبويب Object Privileges، يمكنك من خلال صفحة التبويب هـــذه منـــح امتيازات العناصر للوظيفة الجديـــدة. انظـــر الشكل ٣٧-٣.



الشكل ٣٧-٣

الآن تستطيع منح هذه الوظيفـــة إلـــى مستخدم ما بســـحب هـــذه الوظيفـــة وإفلاتها في عقـــدة Granted Roles المتعلقة بالمستخدم المطلـــوب (انظـــر الشكل ٣٧-٤).



الشكل ٣٧-٤

تعديل الوظائف Altering Roles

تعديل وظيفة باستخدام الأداة Server Manager

يمكنك تعديل وظيفة باستخدام تعليمة ALTER ROLE على الشكل:

ALTER ROLE role
[NOT IDENTIFIED]

[IDENTIFIED BY password | EXTERNALLY | GLOBALLY];

حيث:

- ☆ **role**: اسم الوظيفة التي سيتم تعديلها.
- ☆ **NOT IDENTIFIED**: لتحديد أن المستخدمين الذي منحوا الوظيفة لن يحتاجوا كي يتحقق منهم مخدم أوراكل لتأهيل الوظيفة.
- ☆ **IDENTIFIED**: لتحديد أن المستخدمين الذي منحوا الوظيفة سيحتاجون لكي يتحقق منهم مخدم أوراكل لتأهيل الوظيفة.
- ☆ **BY password**: لتحديد كلمة المرور التي يجب إعطاؤها لتأهيل الوظيفة.
- ☆ **EXTERNALLY**: كي يقوم مخدم أوراكل بالتحقق من وصول المستخدم للوظيفة باستخدام أدوات نظام التشغيل.

تعديل وظيفة باستخدام الأداة Security Manager

تستطيع تعديل وظيفة بالنقر عليها في نافذة Security Manager، تظهر في الجزء الأيمن جميع خصائص هذه الوظيفة، حيث يمكن بسهولة إجراء التعديلات عليها (انظر الشكل ٤-٣٧).

تأهيل وإلغاء تأهيل الوظائف Enabling and Disabling Roles

تأهيل وإلغاء تأهيل وظيفة باستخدام الأداة Server Manager

يمكنك القيام بذلك باستخدام تعليمة SET ROLE على الشكل:

SET ROLE role [IDENTIFIED BY password]
[ALL [EXCEPT role[,role]]]
[NONE]

حيث:

- ☆ **role**: اسم الوظيفة المطلوب تعديلها.

- ☆ **password**: كلمة المرور المتعلقة الوظيفة.
- ☆ **ALL**: لتأهيل جميع وظائف المستخدم عدا تلك المحدّدة بعد EXCEPT.
- ☆ **NONE**: لإلغاء تأهيل الوظيفة المحدّدة.



```
SET ROLE mohib ALL;

SET ROLE mohib NONE;
```

الوظائف وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للوظائف هي:

- ☆ المشاهد المتعلقة بالوظائف: DBA_ROLES,DBA_ROLES_PRIVS, USER_ROLE_PRIVS.
- ☆ المشاهد المتعلقة بالوظائف على الوظائف: ROLE_ROLE_PRIVS.
- ☆ المشاهد المتعلقة بالامتيازات الممنوحة للوظائف: ROLE_SYS_PRIVS, ROLE_TAB_PRIVS.
- ☆ المشاهد المتعلقة بوظائف الدورة: SESSION_ROLE.

قائمة بجميع وظائف القاعدة:

```
SELECT *
FROM sys.dba_roles;
```



قائمة بوظائف الوظائف لمستخدم:

```
SELECT *
FROM sys.dba_role_privs
WHERE grantee='RL_ADMIN_SECU';
```



قائمة بالوظائف الفعالة في الدورة:

```
SELECT *
FROM session_roles;
```





إدارة التشكيلات الجانبية
Administrating Profiles

التشكيلات الجانـــبية عـــبارة عن مجموعة مسمّاة لمقيّدات مصادر محددة يمكن ربطهـــا مـــع مستخدم قاعدة معطيات أوراكل، لذلك فهي تعطينا أداة سهلة لإدارة مقيّدات المصادر.

ونحـــتاج إلى إنشـــاء الـــتشـــكيل الجانـــبي فقط عندما نكون بحاجة لحماية وإدارة قاعدة المعطيات.

إنشاء التشكيلات الجانبية Creating Profiles

يمكــن القيام بذلك باستخدام الأداة Enterprise Manager أو Security Manager أو Server Manager.

إنشاء تشكيل جانبي باستخدام الأداة Server Manager

يمكنك إنشاء تشكيل جانبي باستخدام تعليمة CREATE PROFILE على الشكل:

```
CREATE PROFILE profile LIMIT
[SESSION_PER_USER integer|UNLIMITED|DEFAULT]
[CPU_PER_SESSION integer|UNLIMITED|DEFAULT]
[CPU_PER_CALL integer|UNLIMITED|DEFAULT]
[CONNECT_TIME integer|UNLIMITED|DEFAULT]
[IDLE_TIME integer|UNLIMITED|DEFAULT]
[LOGICAL_READS_PER_SESSION integer|UNLIMITED|DEFAULT]
[LOGICAL_READS_PER_CALL integer|UNLIMITED|DEFAULT]
[COMPOSITE_LIMIT integer|UNLIMITED|DEFAULT]
[PRIVATE_SGA integer[K|M]|UNLIMITED|DEFAULT]
```

حيث:

- ☆ **profile**: اسم التشكيل الجانبي الذي سيتم إنشاؤه.
- ☆ **CPU_PER_SESSION**: زمن المعالج الكلّي مقاساً بواحدة من مئات الثواني لكل دورة.
- ☆ **CPU_PER_CALL**: زمن المعالج الكلّي مقاساً بواحدة من مئات الثواني لكل طلب.
- ☆ **SESSION_PER_USER**: عــدد الــدورات المتزامــنة المســموح به لكل مستخدم.
- ☆ **CONNECT_TIME**: وقت الاتصال المنقضي مقاساً بالدقائق.
- ☆ **IDLE_TIME**: مدّة الوقت غير الفعّال مقاساً بالدقائق.
- ☆ **LOGICAL_READS_PER_SESSION**: عدد كتل المعطيات التي يسمح بقراءتها في كل دورة.

☆ LOGICAL_READS_PER_CALL: عدد كتل المعطيات التـــي يســـمح بقراءتها في كل طلب.

☆ COMPOSITE_LIMIT: لتحديد قيود كلفة المصادر الكلية للدورة.

☆ UNLIMITED: لتحديد أنه يمكن للمستخدم الحصول على مصادر التشــكيلات الجانبية بدون أي قيود.

☆ DEFAULT: لتحديد القيمة الافتراضية لمصادر التشكيلات الجانبية.

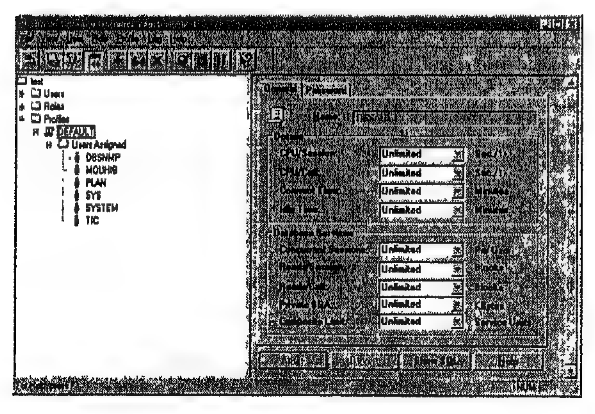
☆ PRIVATE_SGA: المساحة الخاصة في SGA مقاسةً بالبايت.



```
CREATE PROFILE mohib LIMIT
SESSION_PER_USER 5
CPU_PER_CALL UMLIMITED
CONNECT_TIME 10
LOGICAL_READS_PER_CALL 500
IDLE_TIME 60;
```

إنشاء تشكيل جانبي باستخدام الأداة Security Manager

من النافذة الرئيسية Oracle Security Manager، قـم بتوسيع العقـــدة Profiles، تظـــهر قائمـــة التشـــكيلات الجانبيـــة المعرّفـــة، بينـــها التشكيل الجـانبي Default، انظر الشكل ٣٨-١.

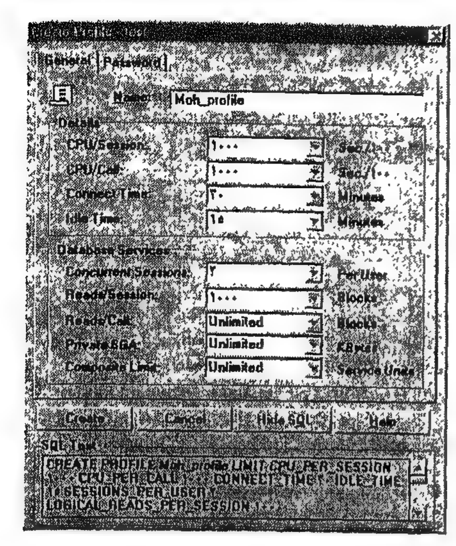


الشكل ٣٨-١

انقر بزر الفأرة الأيمن على العقــدة Profile ثم اختر Create يظهر صندوق حوار مشابه للشكل ٣٨-٢.

تستطيع من خلال هذه النـــافذة تحديــد قيــم مصادر التشكيلات الجانبية كما تمّ شرحها في الفقرة السابقة.

انقــر زر Show SQL لرؤيـــة التعليمـــة الموافقة لإنشاء التشكيل الجانبي.



الشكل ٣٨-٢

تعديل التشكيلات الجانبية Altering Profiles

تعديل تشكيل جانبي باستخدام الأداة Server Manager

يمكنك تعديل تشكيل جانبي باستخدام تعليمة ALTER PROFILE على الشكل:

ALTER PROFILE profile LIMIT
[SESSION_PER_USER integer|UNLIMITED|DEFAULT]
[CPU_PER_SESSION integer|UNLIMITED|DEFAULT]
[CPU_PER_CALL integer|UNLIMITED|DEFAULT]
[CONNECT_TIME integer|UNLIMITED|DEFAULT]
[IDLE_TIME integer|UNLIMITED|DEFAULT]
[LOGICAL_READS_PER_SESSION integer|UNLIMITED|DEFAULT]
[LOGICAL_READS_PER_CALL integer|UNLIMITED|DEFAULT]
[COMPOSITE_LIMIT integer|UNLIMITED|DEFAULT]
[PRIVATE_SGA integer[K|M]|UNLIMITED|DEFAULT]



```
ALTER PROFILE mohib LIMIT
SESSION_PER_USER 8
CONNECT_TIME 20
LOGICAL_READS_PER_CALL 1000
IDLE_TIME 30;
```

يمكنك تعديل قيم التشكيل الجانبي الافتراضي باستخدام تعليمة ALTER PROFILE default مثلاً:

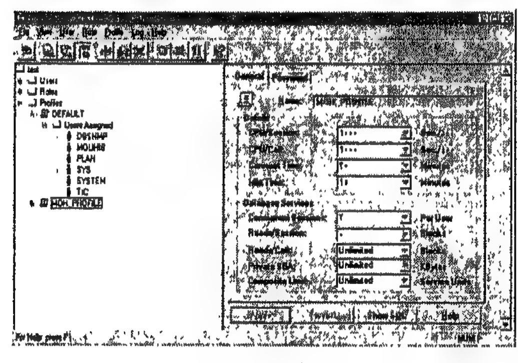


```
ALTER PROFILE default LIMIT
SESSION_PER_USER 10
CPU_PER_CALL 4000
CONNECT_TIME 5
LOGICAL_READS_PER_CALL 1000
IDLE_TIME 20;
```

تعديل تشكيل جانبي باستخدام الأداة *Security Manager*

انقر على التشكيل الجانبي المطلوب تعديله، يظهر في الجزء الأيمن نافذة تحدّد قيم مصادر التشكيل الجانبي المحدد كما في الشكل ٣٨-٣.

تستطيع من خلال هذه النافذة تعديل قيم مصادر التشكيلات الجانبية كما ترغب.



الشكل ٣٨-٣

حذف التشكيلات الجانبية Dropping Profiles

حذف تشكيل جانبي باستخدام الأداة *Server Manager*

يمكنك حذف تشكيل جانبي باستخدام تعليمة DROP PROFILE على الشكل:

```
DROP PROFILE profile [CASCADE]
```

حيث:

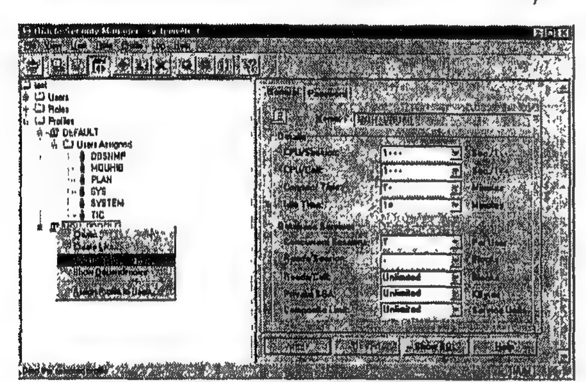
- ☆ **profile**: اسم التشكيل الجانبي المطلوب حذفه.
- ☆ **CASCADE**: لإلغاء التشكيل الجانبي من جميع المستخدمين.



DROP PROFILE mohib CASCADE;

حذف تشكيل جانبي باستخدام الأداة Security Manager

انقر بزر الفأرة الأيمن على التشكيل الجانبي المطلوب حذفه، ثم اختر الأمر Remove كما في الشكل ٣٨-٤.



الشكل ٣٨-٤

ربط التشكيلات الجانبية Assigning Profiles

ربط تشكيل جانبي بمستخدم باستخدام الأداة Server Manager

يمكنك تعديل تشكيل جانبي باستخدام تعليمة ALTER USER على الشكل:

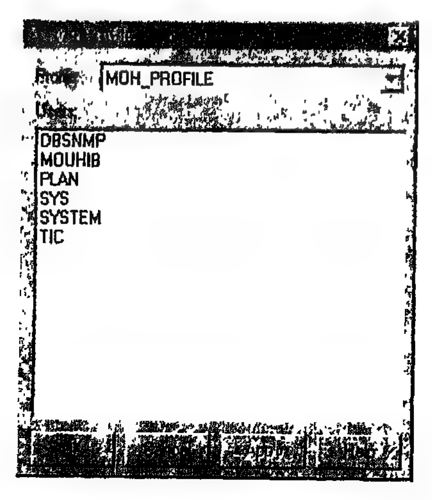
ALTER USER user
PROFILE profile;



```
ALTER USER mohib
PROFILE mohib;
```

ربط تشكيل جانبي باستخدام الأداة Security Manager

يمكن بسهولة ربط تشكيل جانبي بمستخدم أو مخطّـط وذلك بالنقر بزر الفأرة الأيمن على التشـكيل الجانـبي المطلوب واختيار الأمر Assign Profile، يظهـر صـندوق حوار يشـبه الشـكل ٣٨-٥. حـدّد المستخدم أو المخطط وانقر زر Ok.



الشكل ٣٨-٥

التشـكيلات الجانبية وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للتشكيلات الجانبية هي:

☆ المشاهد المتعلقة بالتشكيلات الجانبية ومقيّدات المصادر: DBA_PROFILES, USER_RESOURCE_LIMITS, RESOURCE_COST.



إظهار التشكيلات الجانبية المتعلقة بمدير القاعدة:

```
SELECT * FROM sys.dba_profile
ORDER BY profile;
```



إظهار أوزان المصادر في الدورة الحالية:

```
SELECT *
FROM resource_cost;
```

- إظهار مقيّدات مصادر المستخدم الحالي:



```
SELECT *
FROM user_resource_limits;
```

- إظهار استخدام الذواكر وفق دورات المستخدم:

```
SELECT user_name, value || 'bytes' "Current session
memory"
FROM v$session sess, v$sesstat stat, v$statname name
WHERE sess.sid = stat.sid
      AND stat.statistic# = name.statistic#
      AND name.name = 'session memory';
```




الباب العاشر



٣٩. التصدير.

٤٠. الاستيراد.

٤١. شحن المعطيات.





التصدير Export

تستخدم الأداة Oracle Export مـن أجـل كتابة تعاريف عناصر ومعطيات أوراكـل إلى مـلف ثـنائي خاص، يعرف هذا الملف بملف التصدير Export file ولا يمكن قراءة هذا الملف إلا باستخدام الأداة Oracle Import.

يمكن استخدام الأداة Oracle Export للقيام بمهام عديدة مختلفة:

- للنسـخ الاحتياطي لقاعدة المعطيات Back Up the Database: حيث يمكن تصـدير المعطيات إلى ملف تصدير ومن ثم إعادة استيراد هذه المعطيات إلى قواعد معطيات أخرى مختلفة، لكن يفضل استخدام الأداة Back Up tools.
- لـنقل المعطيـات بيـن قواعد المعطيات حيث يمكن نقل الجداول من قاعدة معطيات إلى أخرى.

❖ لإعادة بناء قاعدة معطيات، خاصةً إذا كانت لديك قاعدة معطيات تحتوي على فضاءات جدولية متناثرة.

❖ تصدير استخدام أدوات الاستيراد والتصدير لإلغاء تناثر قاعدة المعطيات، وهذا يساعد في تحسين الأداء في حال كان التناثر يبطئ عمل القاعدة.

❖ لإعادة تنظيم قاعدة المعطيات: إذا رغبت بإعادة تنظيم مواقع ملفات المعطيات، يمكنك إنشاء فضاءات جدولية جديدة واستخدام الاستيراد Import لإعادة شحن هذه المعطيات.

وكما في أغلب أدوات أوراكل، يمكن إجراء عمليات تصدير المعطيات إما باستخدام الأداة Data Manager أو باستخدام برنامج EXP٨٠.

التصدير باستخدام الأداة Data Manager

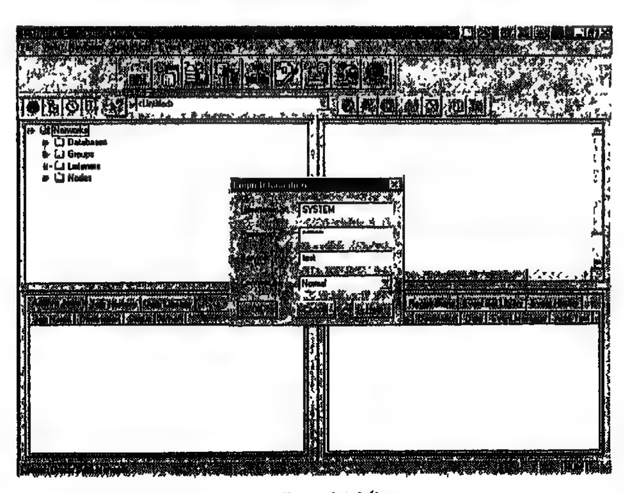
كي تجعل الأداة Data Manager تعمل يجب تشغيل Enterprise Manager, Oracle agent.





*كي تستطيع القيام بتصدير كامل **full export** يجب تشغيل **Data Manager** من قبل مستخدم يمتلك امتيازات **DBA**.*

١- عندما تقوم بتشغيل Data Manager، ستظهر النافذة كما في الشكل ٣٩-١

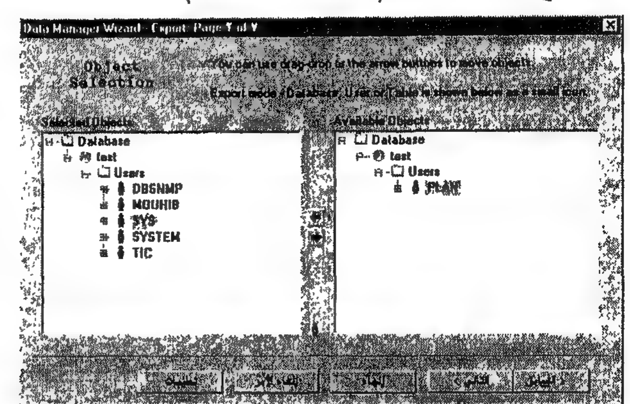


الشكل ٣٩-١

٢-تستطيع الآن القيام بعمليات التصدير باختيار الأمر Export من قائمة Data، يظهر معالج Data Manager Export، تظهر نافذة تطلب تحديد موقع ملف التصدير (انظر الشكل ٣٩-٢).

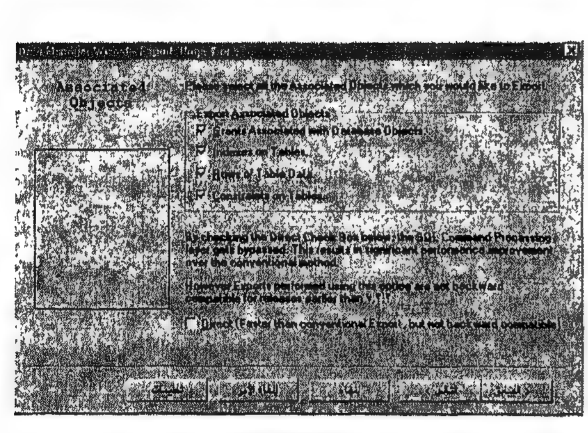
الشكل ٣٩-٢

٣-النافذة الثانية تسمح لك بتحديد العناصر التي تريد تصديرها. افتراضياً يتم تحديد كامل القاعدة. انظر الشكل ٣٩-٣.

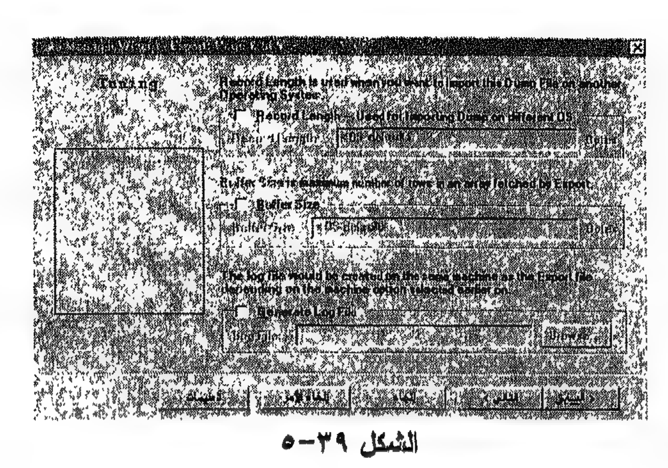


الشكل ٣٩-٣

٤- في النافذة الثالثة، تستطيع اختيار العناصر التي ترغب بتصديرها متضمنة الامتيازات والفهارس وأسطر الجداول والقيود، انظر الشكل ٣٩-٤.

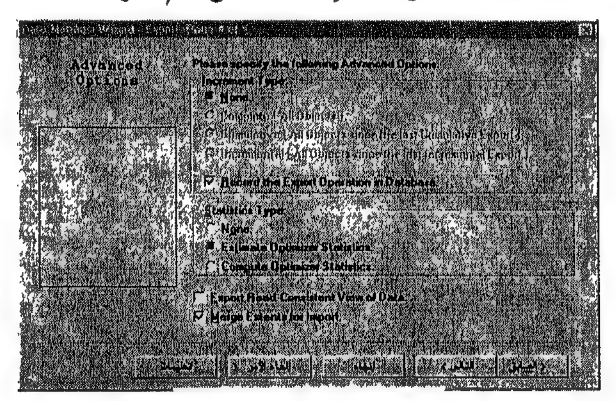


الشكل ٣٩-٤



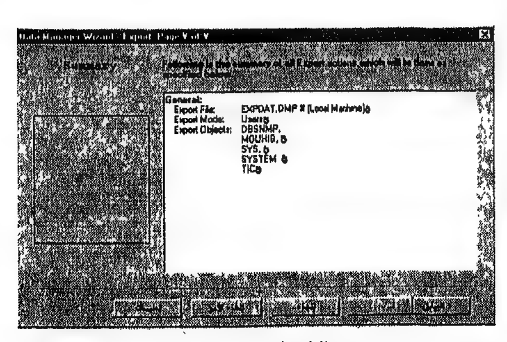
الشكل ٣٩-٥

٥- النافذة الرابعة تسمح لك باختيار طول السجل record length (هذا ضروري إذا أردت نقل ملف التصدير إلى نظام تشغيل آخر)، حجم الذاكرة المؤقتة buffer size وملف الإعادة log file (انظر الشكل ٣٩-٥).

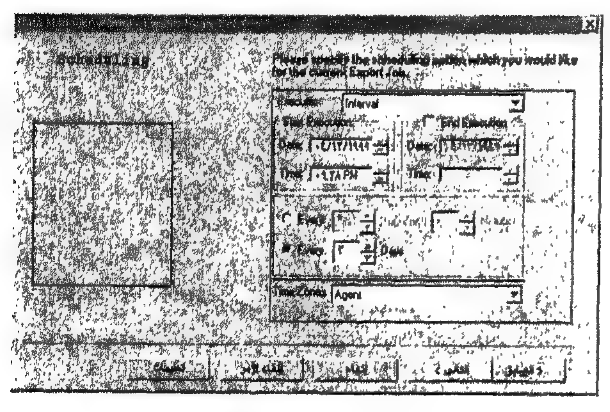


الشكل ٣٩-٦

٦- النافذة الخامسة تسمح لك باختيار إما إجراء تصدير كامل Complete export أو تصدير متزايد incremental export، كما يمكن اختيار أنماط الإحصاءات Statistics modes هذا يحدد فيما إذا كانت الإحصائيات المتوقعة estimated أو المحسوبة Calculated سيتم جمعها للمحلل Optimizer عند استيراد المعطيات. أخيراً تستطيع توصيف نمط التناسق Consistency وكذلك نمط دمج المدى extent merging mode، (انظر الشكل ٣٩-٦).



الشكل ٣٩-٧



الشكل ٣٩-٨

٧- النافذة الأخيرة تلخص عمليات التصدير التي تمّ اختيارها مسبقاً، انقر هنا على زر Finish. انظر الشكل (٣٩-٧).

٨- إذا اخترت جدولة أوقات التصدير Schedule the Export بدلاً من إجرائه مباشرة، ستظهر لك النافذة السادسة كما في الشكل٣٨-٨.

التصدير باستخدام الأداة EXP80

ضمن نظام Windows NT، فإن برنامج التصدير هو EXP80، أما ضمن بقية الأنظمة فإنه يعرف بالاسم EXP.

يمتلك برنامج التصدير عدة أنماط:

- **Full**: لتصدير جميع عناصر قاعدة المعطيات (عدا المخطط SYS).
- **Table**: لتحديد عناصر المخطط المطلوب تصديرها.
- **User**: لتصدير جميع العناصر التي تنتمي إلى مستخدم محدد.

يتم طلب تشغيل هذا البرنامج من خلال سطر الأوامر Command line ووفق الشكل:

EXP 80 username / password [options....]

ويمكن تسهيل استخدام هذا البرنامج وذلك باستخدام ملف وسطاء parameter file يحتوي على وسطاء التصدير حيث يصبح طلب تشغيل برنامج التصدير على الشكل:

EXP 80 username / password PARFILE = filename [options....]

تستخدم وسطاء التصدير الشكل:

PARAMETER = value

يوضح الجدول التالي وسطاء التصدير وشرح عمل كل منها والقيمة الافتراضية:

اسم الوسيط	القيمة التي يأخذها	القيمة الافتراضية	عمل الوسيط
BUFFER	Number	-	يحدد حجم الذاكرة المؤقتة للنسخ Copy buffer (بالبايت) المستخدمة من قبل برنامج التصدير، إذا كانت قيمتها صفر فيتم جلب سطر واحد كل مرة.
COMPRESS	Y or N	Y	يحدد هذا الوسيط كيفية معالجة المدى الابتدائي. فإذا كانت القيمة Y يتم تجميع جميع معطيات الجدول في مدى واحد. وإذا كانت القيمة N يقوم البرنامج باستخدام وسطاء التخزين الحالية لإنشاء مدىً جديد.
CONSISTENT	Y or N	Y	في حال كانت القيمة Y فإن عملية التصدير تكون متماسكة أو منسجمة Consistent لكنها يمكن أن تأخذ حيزاً مهماً من فراغ الاسترجاع ويمكن أن تفشل في حال إجراء تعديلات كثيرة.
CONSTRAINT	Y or N	Y	لتحديد إن كنا نرغب بتصدير قيود الجداول أم لا.
DIRECT	Y or N	N	لتحديد إن كنا نرغب باستخدام خيار التصدير بالمسار المباشر direct path.export، وبالتالي تجاهل طبقة معالجة أوامر SQL ولا يمكن استخدام هذا الخيار مع بعض مخططات العناصر.
FEEDBACK	Number		وذلك لتحديد إن كنا نرغب بإظهار نقطة عند كل عدد محدد بـ number من الأسطر التي تم تصديرها. فإذا كان ١٠٠=

			FEEDBACK فهـذا يعـني أننا سنرى نقطة بعد كل ١٠٠ سطر تم تصديره.
FILE	Filename	EXPDAT.DMP	لتحديد اسم ملف التصدير.
FULL	Y or N	N	لـتحديد إن كنا نرغب بتصدير الفهارس أم لا.
GRANTS	Y or N	Y	لـتحديد إن كنا نرغب بتصدير السماحيات GRANTS أم لا.
HELP	Y or N	N	لإظهار قائمة وسطاء التصدير أم لا.
INCTYPE	Type	–	لتحديد نمط التصدير: COMPLETE أو CUMULATIVE أو INCREMENTAL.
INDEXES	Y or N	Y	لـتحديد إن كنا نرغب بتصدير الفهارس أم لا.
LOG	Log file	–	لـتحديد اسم ملف الإرجاع الذي ستتم فيه كتابة رسائل الخطأ ورسائل الحالة.
OWNER	Owner-name	–	لتحديد قائمة المستخدمين الذين سيتم تصدير العناصر الخاصة بهم.
PARFILE	Para-file	–	لتحديد اسم ملف وسطاء التصدير.
POINT- IN-TIME-RECOVER	Y or N	Y	لـتحديد إن كانت استرجاع Point_ in_ time سيتم إجراؤه عند الاستيراد.
RECORD	Y or N	Y	تحديد أن جداول النظامSYS.INCVID SYS. INCEXP, SYS.INCFIL تسـجل التصـدير الـتجميعي أو التصدير المتزايد.
RECORDLENGTH	Number	–	تحديـد حجـم سجل الملف file record بالبايت، ويستخدم هذا الوسيط في حال نقل ملف التصدير إلى نظام تشغيل مختلف.

RECOVERY_ TABLESPACE	Ts_na me	–	لــتحديد أســماء الفضاءات الجدولية التي تستخدم مع استرجاع point_ in_ time.
ROWS	Y or N	Y	لتحديد إن كان سيتم تصدير معطيات أسطر الجدول أم لا.
STATISTICS	Type	ESTIMAT E	لــتحديد نمــط الإحصــائيات الــناتجة: ESTIMATE أو COMPUTE أو NONE وذلك عند استيراد المعطيات.
TABLES	Tables	–	لــتحديد قائمة الجداول التي سيتم تصديرها عــند اســتخدام برنامج EXPORT بنمط الجداول table mode.
USERID	Userna me / passwo rd	–	لــتحديد اســم وكــلمة المــرور الخاصة بالمســتخدم الــذي ســيقوم بإجراء عملية التصدير.



كمثال على ملف وسطاء التصدير:

FULL= Y
BUFFER= 8192
FILE= D:/ database/ export / EXPDAT. DMP
STATISTICS= COMPUTE

وفــي حال لم تحدد ملف الوسطاء سيطلب منك برنامج التصدير تحديد قيمة كل وسيط.

التصدير وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للتصدير هي:

☆ المشــــاهد المتعـــــلقة بالتصـــــدير: DBA_EXP_FILES, DBA_EXP_OBJECTS, DBA_EXP_VERSION.





الاستيراد Import

تستخدم هذه الأداة من أجل شحن المعطيات التي تم تصديرها إلى قاعدة المعطيات باستخدام برنامج التصدير EXPORT.

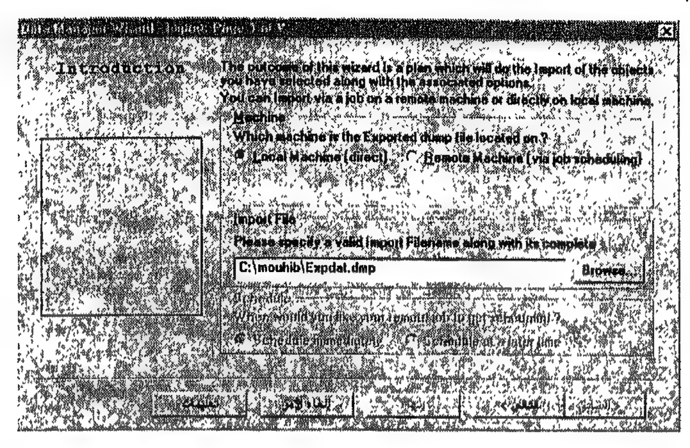
ولا يمكن لأداة الاستيراد سوى قراءة المعطيات التي تمّ تصديرها.



*عندما تحتاج إلى شحن معطيات أخرى إلى قاعدة معطيات أوراكل، يمكنك استخدام الأداة SQL*Loader وسنقوم بشرحها في الفصل ٤١.*

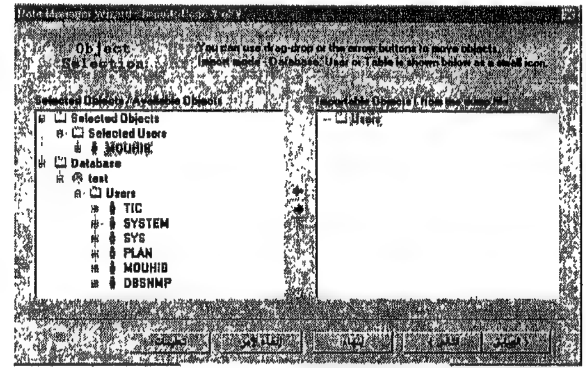
الاستيراد باستخدام الأداة Data Manager

١-في نافذة Data Manager اختر الأمر Import من قائمة Data. سيقوم معالج خاص بإظهار نافذة تسمح لك بتحديد اسم ملف التصدير export file الذي ترغب باستيراده. انظر الشكل٤٠-١.



الشكل ٤٠-١

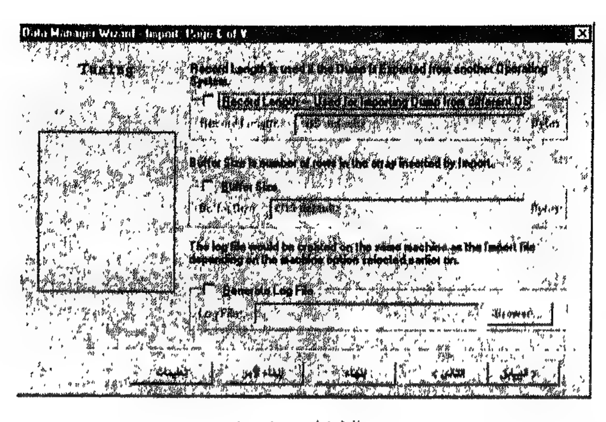
٢-في النافذة الثانية اختر العناصر التي ترغب باستيرادها، انظر الشكل ٤٠-٢.



الشكل ٤٠-٢

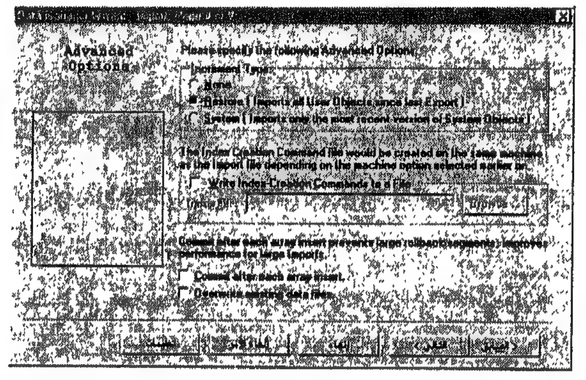
٣-في النافذة الثالثة، تظهر العناصر التي تم تصديرها، يمكنك هنا إلغاء استيراد أي منها. انظر الشكل ٤٠-٣.

الشكل ٤٠-٣



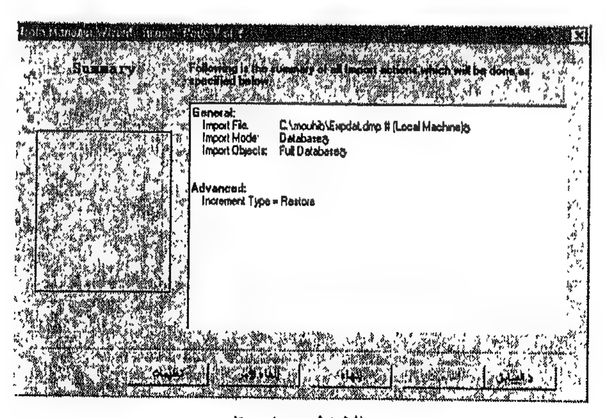
الشكل ٤٠-٤

٤-تسمح لك النافذة الرابعة باختيار طول السجل (في حال استيراد ملفات من نظم تشغيل أخرى)، وحجم الذاكرة المؤقتة وملف الإعادة. (يفضل هنا ترك القيم الافتراضية). انظر الشكل ٤٠-٤.



الشكل ٤٠-٥

٥-تحتوي النافذة الخامسة على خيارات متقدمة كنمط الاستيراد وكذلك فيما إذا كنت ترغب باستيراد كافة المعطيات أو المعطيات الجديدة فقط. يمكنك أيضاً كتابة تعليمة إنشاء فهرس على ملف بدلاً من استيراده. انظر الشكل ٤٠-٥.

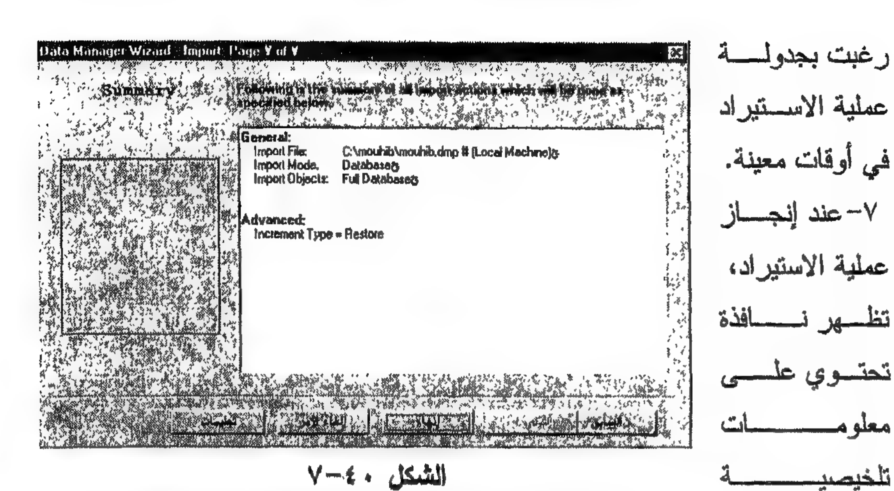


الشكل ٤٠-٦

٦-النافذة الأخيرة تظهر لك قائمة بالخيارات التي تم تحديدها من أجل إنجاز عملية الاستيراد (انظر الشكل ٤٠-٦). وتظهر لك نافذة سادسة فقط في حال

رغبت بجدولة عملية الاستيراد في أوقات معينة.

٧- عند إنجاز عملية الاستيراد، تظهر نافذة تحتوي على معلومات تلخيصية ومعلومات الحالة (انظر الشكل ٤٠-٧).



الشكل ٤٠-٧

الاستيراد باستخدام الأداة IMP80

تستخدم هذه الأداة بنمط أوامر السطر Command_ line وتأخذ الاسم IMP80 في نظام Windows NT والاسم IMP في بقية الأنظمة.

يتم طلب تشغيل هذا البرنامج من خلال سطر الأوامر Command line ووفق الشكل:

IMP80 username / password [options....]

وتتشابه خيارات هذا البرنامج مع خيارات برنامج EXP80 إلى حد كبير.





شحن المعطيات

Data Loading

تعتبر الأداة SQL*Loader إحدى أدوات أوراكل المستخدمة في شحن المعطيات إلى قاعدة معطيات أوراكل. وبينما صممت الأداة Import لقبول معطيات بنمط محدد، فإن SQL* Loader مصمم بشكل مرن لقبول المعطيات بأنماط عديدة.

شحن المعطيات باستخدام الأداة SQL*LOADER

يتعامل SQL* Loader مع نمطين من أنماط الملفات: ملف معطيات الدخل وملف تحكم الشحن يستخدم ملف التحكم بتحديد تنسيق ملف المعطيات وتحديد معلومات أخرى كأنماط الأعمدة وفواصل الحقول وغيرها. ويمكن أن يحتوي ملف التحكم على المعطيات نفسها.

يمكن أن يحتوي ملف التحكم على أسطر متعددة لكل تعليمة، ولا يهم نمط الأحرف إلا إذا كانت بين فاصلتين علويتين.

ويتعامل ملف التحكم مع أكثر من ٩٠ كلمة مفتاح يمكن استخدامها لتحديد تنسيق المعطيات كذلك كيفية شحن هذه المعطيات. وهو يحتوي على أوامر تقوم بالأعمال الأساسية التالية:

- ❖ ما هي العملية المطلوب إنجازها.
- ❖ مكان إيجاد ملف معطيات الدخل.؟
- ❖ تنسيق المعطيات.

ويمكن أن تكون المعطيات بطول ثابت أو غير محدد.

يوضح المثال التالي، ملف تحكم بطول سجل ثابت:

```
LOAD DATA
INFILE 'D:/database/load/Moh1.Dat'
INTO TABLE ''MOUHIB'.Moh1 (
        Id POSITION ( 01: 02 ) INTEGER EXTRNAL,
        Name POSITION ( 05: 08 ) CHAR,
        Age POSITION ( 11: 11 ) INTEGER EXTERNAL )
```

أما المثال التالي فيوضح ملف تحكم بطول سجل متغير:



```
LOAD DATA
INFILE 'D: /database/load/Moh2.Dat'
INTO TABLE ''MOUHIB''.Moh2
FILEDS TERMINATED BY '','' OPTIONALLY
ENCLSED BY ' \\ '
(id, name CHAR, owner_id)
```

توجد مجموعة من الخيارات المفيدة المتاحة للأداة SQL Loader أهمها:

- ❖ **BAD= filename**: اسم الملف الذي ستوضع فيه المعطيات السيئة.

- ❖ CONTROL= filename: اسم ملف التحكم.
- ❖ DATA= filename: اسم ملف المعطيات.
- ❖ DIRECT= (TRUE or FALSE): لتحديد إن كان سيتم استخدام مسار الشحن المباشر direct path loader.
- ❖ PARALLEL = (TRUE or FALSE): لتحديد إن كان سيتم استخدام طريقة الشحن المتوازي أم لا.
- ❖ LOAD= n: عدد السجلات التي سيتم شحنها.
- ❖ LOG= filename: اسم ملف الإرجاع.
- ❖ ROWS= n: عدد الأسطر في كل مصفوفة إدراج.
- ❖ SKIP = n: عدد السجلات المنطقية التي سيتم تجاوزها.
- ❖ UNRECOVERABLE= (TRUE or FALSE): يستخدم هذا الخيار فقط مع مسار الشحن المباشر ويحدد إن كان سيتم حفظ عمليات شحن المعطيات في ملفات الإرجاع أم لا.

استخدام مسار الشحن المباشر DIRECT PATH LOADER

تستخدم تقنية الشحن التقليدية تعليمات INSERT لشحن المعطيات إلى قاعدة المعطيات، لذلك فهي تمر بجميع الخطوات المطلوبة لإنجاز تعليمة INSERT هذه.

ومن أجل تحسين الأداء يمكن استخدام مسار الشحن المباشر direct path loader والذي يقوم بإدراج المعطيات مباشرة في ملفات المعطيات دون المرور بالكثير من العمليات المنطقية المطلوبة عند إجراء الشحن بطريقة تقليدية.

وعلى الرغم من أن مسار الشحن المباشر أسرع وأكثر فعالية، إلا أنه توجد بعض القيود أهمها:

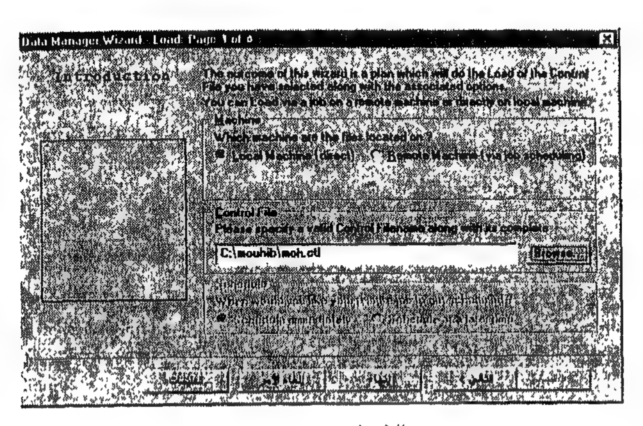
- ❖ لا يمكن استخدامه مع الجداول التجمّعيّة clustered table.
- ❖ لا يمكن استخدامه مع الجداول في حال كانت التحويلات فعالة.

ومن الحسنات التي يمكن ملاحظتها على هذه الطريقة هي إمكانية استخدام خيار عدم الاسترداد UNRECOVERABLE مما يسرع عملية الشحن.

شحن المعطيات باستخدام الأداة Data Manager

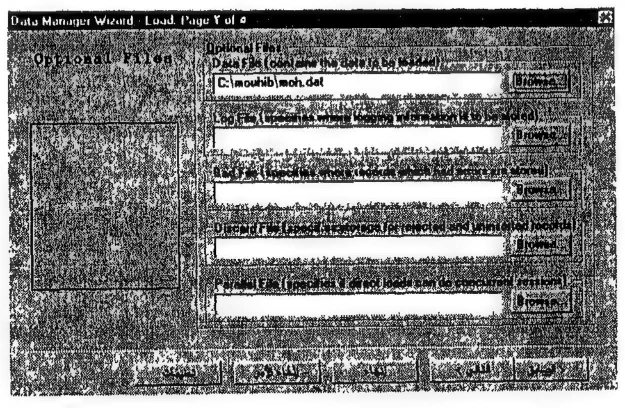
من النافذة الرئيسية للبرنامج Data Manager اطلب الخيار Load من قائمة Data، فيتم تشغيل معالج شحن المعطيات.

١- النافذة الأولى، تطلب منك تحديد ملف التحكم Control file، انظر الشكل ٤١-١.



الشكل ٤١-١

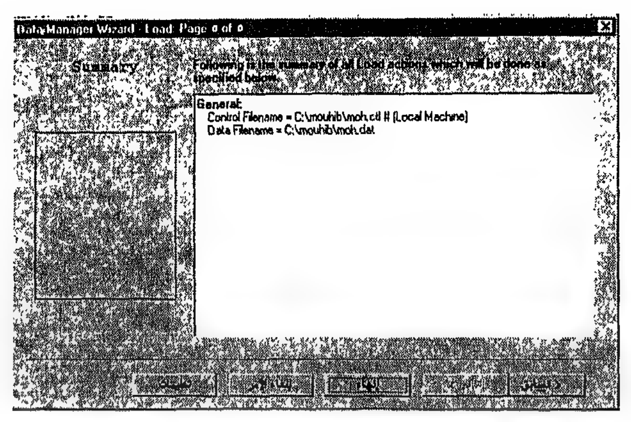
٢- أما النافذة الثانية فتطلب منك تحديد اسم ملف المعطيات Data File وملف الإرجاع Log File وملف الأخطاء Bad File وملف السجلات التي لم تشحن Discard File وملف عمليات الشحن المتزامنة Parallel File، انظر الشكل ٤٢-٢.



الشكل ٤١-٢

٣-أما النافذة الثالثة فتسمح لك باختيار خيارات متقدمة، كمسار الشحن المباشر، والسجلات التي سيتم تجاوزها Skipped records والسجلات التي سيتم شحنها records to load، والسجلات التي سيتم تسجيلها Rows Per Commit والعدد الأعظم لأخطاء الإدراج التي تنهي عملية الشحن Maximum insert Errors for Load Termination، والعدد الأعظمي للسجلات التي لم تشحن والتي لم تنه عملية الشحن Maximum Discard Records For Load Termination وكذلك العدد الأعظم للمصفوفات المندمجة بالبايت Maximum Bind Array in Bytes (انظر الشكل ٤١-٣)، بعد ذلك تظهر لك نافذة تلخيصية عن خيارات عملية الشحن (انظر الشكل ٤١-٤).

الشكل ٤١-٣



الشكل ٤١-٤

شحن المعطيات باستخدام الأداة SQLLDR٨٠

يمكن تشغيل SQL* Loader من سطر الأوامر باستخدام البرنامج SQLLDR٨٠ (طبعاً ضمن نظام Windows NT، أما في بقية الأنظمة فيأخذ الاسم SQLLDR).

تأخذ هذه التعليمة الشكل:

يتم طلب تشغيل هذا البرنامج من خلال سطر الأوامر Command line ووفق الشكل:

SQLLDR80 username / password [options… .]

وعـــلى اعتبار أن معظم الخيارات يتم استخدامها في ملف التحكم فإنه توجد خيارات قليلة مستخدمة ضمن سطر الأوامر أهمها:

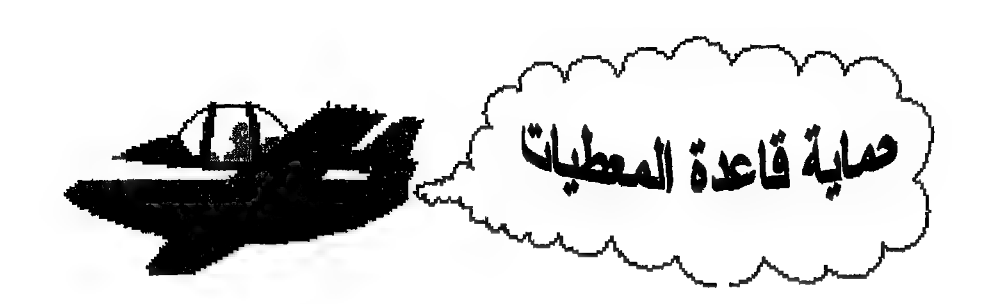
- **USERNAME= username / password**: لتحديد اسم المستخدم الذي سنقوم بتشغيل برنامج الشحن ضمنه.
- **PARFILE = Parameter_ file**: اخـــتياري لتحديد اسم ملف الوسطاء الذي يحتوي على وسطاء إضافية.
- **BAD=filename**: لتحديد اسم ملف الأخطاء.
- **Control= file**: اسم ملف التحكم.

شحن المعطيات وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة لشحن المعطيات هي:

- المشاهد المتعلقة بالإحصائيات عن شحن المعطيات: V$LOADCSTAT, V$LOADTSTAT.


الباب الحادي عشر



٤٢ . حماية المعطيات.

٤٣ . النسخ الاحتياطي.

٤٤ . الاسترداد.





حماية المعطيات
Data Protection

عندما تقـوم بإنشـاء تطـبيقاتك فالمهمّة الأساسية التي يجب عليك أخذها بعين الاعتـبار هي حماية معطياتك الضرورية من أيّ نمط من المشاكل ابتداءً مـن الأعطـال البسيطة الناتجة عن توقّف مفاجئ للتيار الكهربائي وحتى الأعطال الكبيرة كتعطّل أقراص التخزين مثلاً.

سـنقوم هـنا بشـرح كيفيـة استخدام تقنيات النسخ الاحتياطي والاسترداد من أجل حماية وصيانة قاعدة المعطيات.

لكن ماهي المشاكل التي يمكن أن تواجهك؟

هنالك مشاكل كبيرة يمكن أن تواجهك لعلّ أكثرها أعطال النظام كتوقف الطاقة المفاجئ أو لعلّة في البرمجيات أو لإخفاق إجراءات نظام التشغيل.

طبعاً هذا النوع من الأعطال قد يؤدي إلى إخفاق إجراءات الخلفية الخاصّة بأوراكل مما يسبّب توقف مخدّم قاعدة المعطيات أو مايسمى بتعطل هيئة قاعدة المعطيات database instance crash. بشكل عام فإنّ تعطّل هيئة قاعدة المعطيات لا يسبّب تعطّل البيئة الفيزيائية لقاعدة المعطيات، وإنما يسبّب ضياع جميع العمليات والمعطيات الموجودة في ذاكرة الهيئة لحظة حدوث العطل إلا إذا كتبت هذه المعطيات على القرص من قبل.

نمط آخر من المشاكل التي يمكن أن تواجهك هو ضياع أحد ملفات المعطيات بسبب خطأ المسؤول عن القاعدة أو فساد الملف أو حصول عطل على القرص.

هذا النمط من المشاكل هو أحد المشاكل الخطيرة التي يجب أن تحضّر نفسك لها.

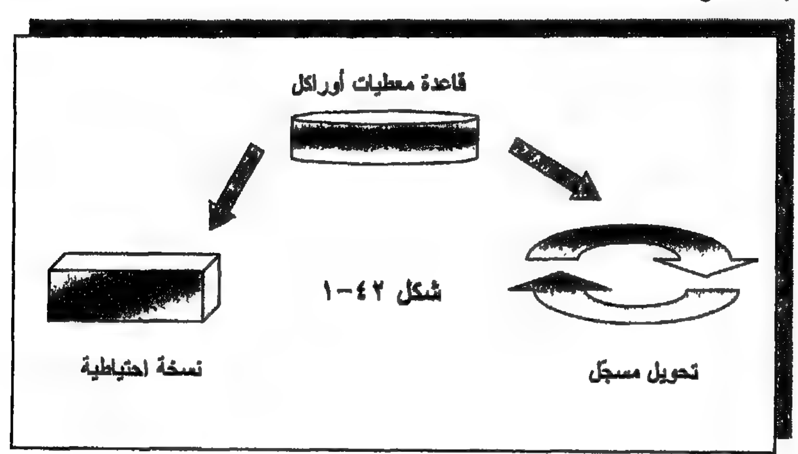
لنفترض مثلاً أنك كنت تعمل خارج أوقات الدوام وتعبت كثيراً وقمت بشكل غير مقصود بحذف أحد ملفات المعطيات. في هذه الحالة لن يكون بإمكانك استرداد ملفك هذا إلا إذا كانت لديك نسخة احتياطية من قاعدة المعطيات تحتوي على ملف المعطيات المحذوف، وإذا رغبت باسترداد جميع الأعمال المثبتة منذ آخر عملية نسخ احتياطي، فيجب عليك امتلاك مجموعات إرجاع الحركات التي تمّ إجراؤها.

بالإضافة إلى ذلك فإن العديد من الشركات التي تستخدم أوراكل لتنفيذ معظم أعمالها، تحتاج إلى تجهيزات احتياطية خاصة تؤمّن لها استمرارية العمل في حال حدوث مشكلة خطيرة أو كارثة ما، لذلك تؤمن لك أوراكل إمكانية استخدام تقنية قاعدة المعطيات المستعدّة للعمل Standby database والتي تؤمن لك الحماية الضرورية.

نظرة على تقنيات حماية قاعدة المعطيات

قبل أن نقوم بشرح التقنيات العديدة لحماية قاعدة معطيات لنأخذ فكرة سريعة عن هذه التقنيات وكيفية عملها.

يوضح الشكل ٤٢-١ الطرق الأساسية لحماية قاعدة المعطيات وهي النسخ الاحتياطي لقاعدة المعطيات database backups وتسجيل التحويلات transaction log، فعملية النسخ الاحتياطي النظامية والمتكررة تجعل بالإمكان استرجاع الملفات الضائعة، إما لخطأ المستخدم أو لعطب الملف أو تعطل القرص. أما تقنية تسجيل التحويلات فهي عبارة عن مجموعة من ملفات نظام التشغيل التي تقوم بتسجيل التغييرات الحاصلة على قاعدة المعطيات نتيجة التحويلات المثبتة. وخلال عملية تثبيت تحويل، يقوم أوراكل بكتابة المعلومات اللازمة في ملف الإرجاع لإعادة العمل الذي تمّ إجراؤه على القاعدة خلال عملية الاسترداد.



شكل ٤٢-١

يحتوي Oracle٨ على الأداة Recover Manager التي تقوم بتبسيط وأتمتة عملية النسخ الاحتياطي والاسترداد من خلال الاحتفاظ بأثر لمعلومات النسخ الاحتياطي ضمن ما يسمى بمجلد الاسترداد recovery catalog، وهو عبارة عن مجموعة جداول ومشاهد قاعدة المعطيات التي تقوم الأداة Recovery Manager باستخدامها لتسجيل معلومات خاصة حول كل عملية نسخ احتياطي يتم إجراؤها.

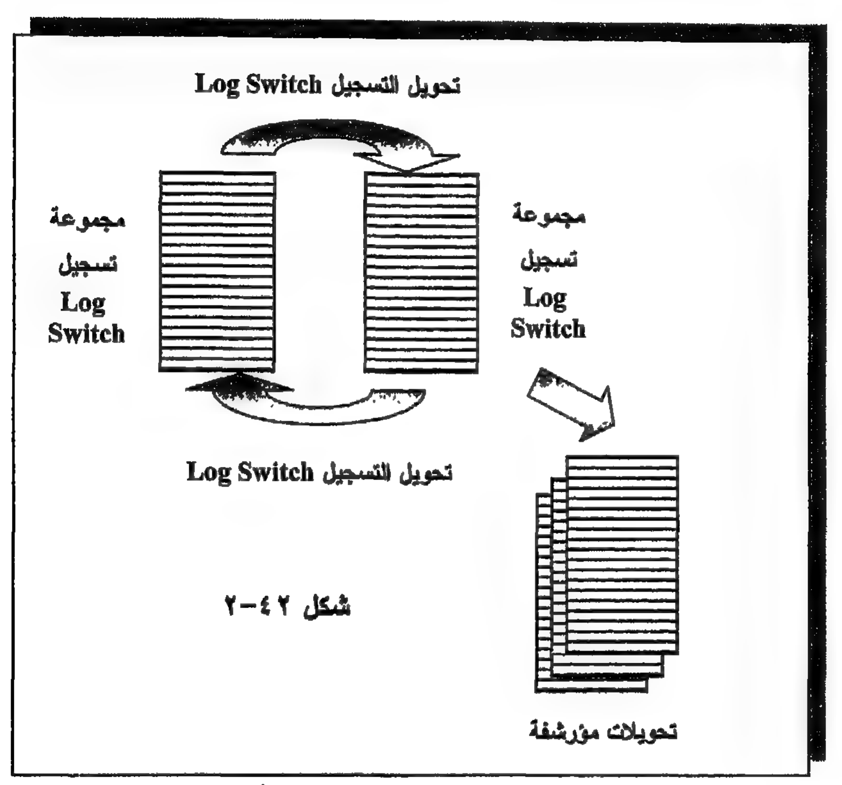
ويتم استخدام المعلومات الموجودة في هذا المجلد لاسترداد قاعدة المعطيات المعطوبة. طبعاً يجب عدم إنشاء مجلد الاسترداد في نفس قاعدة المعطيات التي تقوم بحمايتها.

تسجيل التحويلات Transaction log

ويسمى أيضاً تسجيل الإرجاع Redo log، وهي عبارة عن جزء هام من قاعدة المعطيات لحماية عمل جميع التحويلات المثبتة.

ويتم ذلك بالتسجيل الفوري للتغييرات التي تحدث بالتحويلات المثبتة. وبالتالي فإنه في حال عطل ما، يستطيع أوراكل قراءة تسجيلات التحويلات الجارية خلال عملية الاسترداد وإعادة عمل التحويلات المثبتة المفقودة من قاعدة المعطيات.

ويتم ذلك من خلال مجموعات من ملفات التسجيل، حيث تتكون كل مجموعة من ملفين أو أكثر تحتوي على مداخل تسجيلات التحويلات المثبتة (انظر الشكل ٤٢-٢).



شكل ٤٢-٢

وخلال عملية إقلاع المخدم، يقوم الإجراء LGWR باختيار أحد هذه المجموعات وكتابة التغييرات عليها وعندما تمتلئ هذه المجموعة، يقوم أوراكل بعملية تحويل التسجيل Log Switch وذلك بإغلاق المجموعة الأولى والانتقال إلى مجموعة أخرى واستكمال كتابة التغييرات.

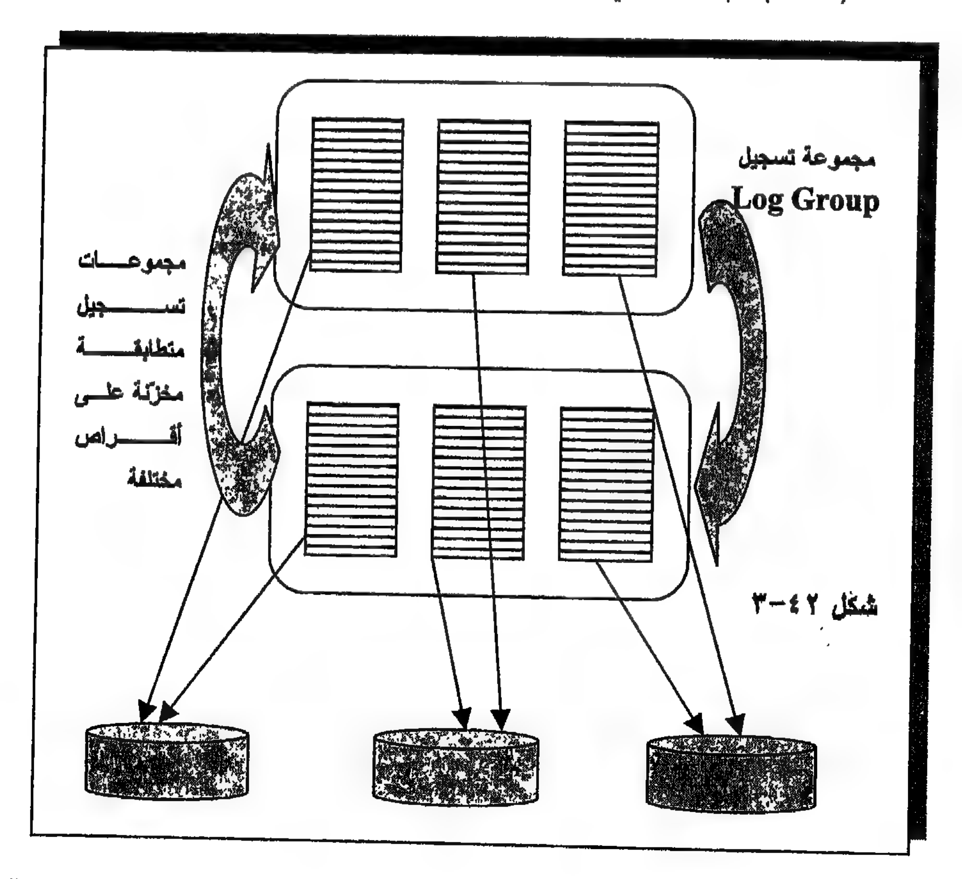
يمكن اختيار أرشفة مجموعات تسجيل التحويلات وذلك من أجل كتابة محتوى تسجيل التحويلات في ملفات عند كل عملية تسجيل التحويل من مجموعة إلى أخرى، وهنالك إجراء خلفية مسؤول بشكل دائم عن أرشفة تسجيلات التحويلات هو الإجراء ARCH،

ويقوم أوراكل بتسمية كل مجموعة تسجيل مؤرشفة برقم تسلسل تسجيل وحيد Log sequence number.

عملية الأرشفة هذه توفّر حماية كاملة لقاعدة المعطيات من الأعطال الكبيرة التي تصيب التجهيزات وبخاصّة أقراص التخزين، ويفضّل إنشاء نسخ احتياطية من هذه الملفات المؤرشفة على أقراص تخزين غير فعّالة كالأشرطة الممغنطة Tapes.

لاحظ أنه عندما يصل أوراكل إلى آخر مجموعة من مجموعات تسجيل التحويلات وينتهي من تعبئتها يعيد الكتابة من جديد على أول مجموعة وذلك فوق ملفات هذه المجموعة، لذلك إذا لم تتم أرشفة تسجيل التحويلات بشكل مستمر فإنّ العديد من التحويلات في هذه الحالة ستضيع عليك.

يمكن أيضاً مضاعفة تسجيل التحويلات بإنشاء مجموعات تسجيل وفي كل مجموعة عدة ملفات (أعضاء) يتم وضعها في عدّة أقراص تخزين وذلك كما في الشكل ٤٢-٣:

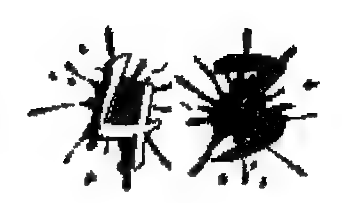


شكل ٤٢-٣

لاحظ أنه يتم تناسخ أعضاء مجموعات التسجيل بحيث يقوم الإجراء LGWR بالكتابة بشكل متزامن على جميع أعضاء كل مجموعة، فإذا تعطّل أيّ من هذه الأعضاء يمكن للإجراء السابق متابعة الكتابة على بقية أعضاء المجموعة.
وكما ذكرنا سابقاً فإنّ الإجراء CKPT مسؤول عن إجراء نقاط التدقيق وذلك خلال كل عملية تسجيل تحويل أو بشروط ومجالات محدّدة. وخلال نقاط التدقيق يقوم الإجراء DBWR بكتابة جميع محتويات كتل المعطيات المعدّلة إلى ملفات المعطيات، ويمكن لهذه النقاط تحديد كميّة مداخل تسجيلات الإرجاع التي يجب على أوراكل تطبيقها عند حدوث عطل في النظام.

حماية ملفات التحكّم

كما نعرف فإنّ ملف التحكّم يحتوي على توصيف للبنية الفيزيائية لقاعدة المعطيات، كذلك فهو يحتوي على معلومات عن نقاط التدقيق وتسجيل التحويلات لإدارة أنماط مختلفة من عمليات استرداد القاعدة، بالإضافة إلى ذلك فإنّ النسخة الجديدة من أوراكل Oracle8 تمكّن من استخدام ملف التحكّم لتسجيل المعلومات عن النسخ الاحتياطي لقاعدة المعطيات.
لذلك فإنّ ملف التحكم مهم جداً ولايمكن لقاعدة المعطيات العمل بشكل سليم بدون هذا الملف.
ولكي تقوم بحماية ملف التحكّم في قاعدة معطياتك، يفضّل أن تقوم بمضاعفته في عدة مواقع بحيث يقوم أوراكل بالتعديل في جميع النسخ في الوقت نفسه. وعندما يتعطّل أحد هذه الملفات، يمكن لقاعدة المعطيات استخدام بقية الملفات دون حدوث أي توقف.





النسخ الاحتياطي Backup

النسخة الاحتياطية لقاعدة المعطيات عبارة عن مجموعة من الملفات التي تكـــوّن قاعدة المعطيات، فإذا حدث عطل أو ضياع لأحد ملفات القاعدة، يمكنـــك جلب نسخة من هذا الملف من النسخة الاحتياطية وإرجاعه إلى القاعدة.

توجد أنماط عديدة للنسخ الاحتياطي أهمها:

* **النسخ الاحتياطي لكامل قاعدة المعطيات Whole Database backups**: وهـو يتضمن نسخة من جميع ملفات معطيات القاعدة، إضافةً إلى ملفات التحكـــم ويمكـــن إجراء إما نسخ احتياطي لقاعدة معطيات مفتوحة أو لقاعدة معطيات مغلقة.

ويسمى النسخ الاحتياطي لقاعدة المعطيات المفتوحة بالنسخ الاحتيـــاطي غـــير المتناســـق Inconsistent Backup كون المعطيات الموجودة في ملفات القاعدة المفتوحة يمكـــن أن

يتم تعديلها من خلال التحويلات أثناء عملية النسخ الاحتياطي، لذلك يجب استخدام نمط ARCHIVELOG كي يستطيع أوراكل إعادة توليد التحويلات المفقودة.

أما النسخ الاحتياطي لقاعدة المعطيات المغلقة فيسمى بالنسخ الاحتياطي المتناسق Consistent Backup، ويجب استخدام نمط NOARCHIVELOG لعدم وجود تحويلات مفقودة في هذه الحالة.

* **النسخ الاحتياطي للفضاءات الجدولية Tablespace Backups:** يمكنك ضمن أوراكل إجراء نسخ احتياطي للفضاءات الجدولية بشكل منفصل، حيث يتمّ إجراء النسخ الاحتياطي للملفّات المكونّة للفضاء الجدولي المحدد.

يفيد هذه النوع بإجراء النسخ الاحتياطي لجزء من قاعدة المعطيات تقوم التطبيقات بإجراء التعديلات عليه أكثر من غيره.

هنا يوجد أيضاً نوعان هما النسخ الاحتياطي للفضاءات الجدولية الفعّالة والنسخ الاحتياطي للفضاءات الجدولية غير الفعّالة.

* **النسخ الاحتياطي لملف معطيات وحيد Data File Backup.**

* **مجموعة النسخ الاحتياطي Backup Set:** يتم إنشاؤها باستخدام مدير الاسترداد Recovery Manager وهي تحتوي على عدّة ملفات من نفس النمط، إما ملفات معطيات أو ملفات تسجيل إرجاع مؤرشفة أو ملفات تحكّم.

ويمكن استخدام إما مجموعة نسخ احتياطية كاملة Full Backup Set، وتتضمن جميع الكتل الموجودة في جميع ملفات المعطيات في مجموعة النسخ الاحتياطي، أو مجموعة النسخ الاحتياطي المتزايدة Incremental Backup Set، وتتضمن فقط الكتل التي تمّ تعديلها في ملفات المعطيات اعتباراً من آخر عملية نسخ احتياطي تمّ إجراؤها.

* **النسخ الاحتياطي المنطقي Logical backup:** وذلك باستخدام أداة التصدير Export وهو يتضمّن جميع مخططات العناصر في القاعدة أو في مخطط محدّد أو حتى في جدول وحيد.

تدعى مجموعة النسخ الاحتياطي الكاملة بالنسخ الاحتياطي بالمستوى0، (level 0 back up)، أما مجموعة النسخ الاحتياطي المتزايدة فيمكن أن تأخذ8 مستويات ، فالنسخ الاحتياطي بالمستوىn يتضمن جميع الكتل من مجموعة النسخ الاحتياطي التي تم تعديلها من آخر عملية نسخ احتياطي بالمستوىn أو أقل.

لنفترض مثلا أنك قمت بإجراء عملية نسخ احتياطي لمجموعة نسخ احتياطي لملفات معطيات يوم السبت بمستوى0 من النسخ الاحتياطي.

- ☆ يوم الأحد، قمت بإجراء نسخ احتياطي بالمستوى2 لمجموعة النسخ الاحتياطي لملفات المعطيات. مجموعة النسخ الاحتياطي هذه تتضمن فقط الكتل التي تم تعديلها من آخر عملية نسخ احتياطي بالمستوى0 التي تم إجراؤها يوم السبت.
- ☆ ويوم الاثنين قمت بعملية نسخ احتياطي أخرى بالمستوى2 لمجموعة النسخ الاحتياطي لملفات المعطيات، مجموعة النسخ الاحتياطي تتضمن فقط الكتل التي تم تعديلها منذ عملية النسخ الاحتياطي بالمستوى2 التي تم إجراؤها يوم الأحد.
- ☆ يوم الثلاثاء قمت بإجراء عملية نسخ احتياطي بالمستوى 1 لمجموعة النسخ الاحتياطي لملفات المعطيات. مجموعة النسخ الاحتياطي تتضمن فقط الكتل التي تم تعديلها منذ عملية النسخ الاحتياطي بالمستوى0 التي تم إجراؤها يوم السبت.
- ☆ يوم الأربعاء قمت بإجراء عملية نسخ احتياطي بالمستوى2 لمجموعة النسخ الاحتياطي لملفات المعطيات. مجموعة النسخ الاحتياطي تتضمن فقط الكتل التي تم تعديلها منذ أخر عملية نسخ احتياطي بالمستوى1 التي تم إجراؤها يوم الثلاثاء.
- ☆ يوم الخميس قمت بإجراء عملية نسخ احتياطي بالمستوى2 لمجموعة النسخ الاحتياطي لملفات المعطيات مجموعة النسخ الاحتياطي تتضمن فقط الكتل التي تم تعديلها منذ آخر عملية نسخ احتياطي بالمستوى2 تم إجراؤها يوم الأربعاء.
- ☆ يوم الجمعة قمت بإجراء عملية نسخ احتياطي بالمستوى1 لمجموعة النسخ الاحتياطي لملفات المعطيات مجموعة النسخ الاحتياطي تتضمن فقط الكتل التي تم تعديلها منذ آخر عملية نسخ احتياطي بالمستوى1 تم إجراؤها يوم الثلاثاء.
- ☆ يوم السبت قمت بإجراء عملية نسخ احتياطي بالمستوى0 لمجموعة النسخ الاحتياطي لملفات المعطيات، وهكذا ...

يوضح المثال السابق بأن أغلب استراتيجيات النسخ الاحتياطي تتضمن دمجا لمجموعة النسخ الاحتياطي لملفات المعطيات المتزايدة والكاملة.

يمكن إجراء النسخ الاحتياطي باستخدام العديد من الأدوات مثل Server Manager و Oracle Backup Manager و Backup Manager.

النسخ الاحتياطي باستخدام الأداة Server Manager

كما ذكرنا فإنّ عملية النسخ الاحتياطي هامة جداً لحماية المعطيات، لذلك يجب إجراء نسخ احتياطي لكامل قاعدة المعطيات عند إنشائها، كذلك عند إجراء تغييرات هامة على بنيتها كإضافة ملفات جديدة أو فضاءات جدولية وغيرها.

طبعاً يجب إجراء نسخ احتياطي لجميع ملفات القاعدة كملفات المعطيات وملفات الإرجاع وملفات التحكّم.

النسخ الاحتياطي لملفّات المعطيات في النمط غير المؤرشف NOARCHIVELOG

يمكنك معرفة أسماء ملفات المعطيات في القاعدة بكتابة التعليمة:

*SELECT * FROM dba_data_files;*

أما لمعرفة أسماء ملفات التحكّم فنكتب التعليمة:

*SELECT * FROM v$parameter*
WHERE name LIKE 'control_files';

وأخيراً لمعرفة ملفات الإرجاع:

*SELECT * FROM v$logfile;*

لبدء عملية النسخ الاحتياطي، قم بإطفاء القاعدة بالطريقة العادية Normal أو Immediate. ثم استخدم بعدها أحد برامج النسخ الاحتياطي المتوفرة في نظام التشغيل لحفظ الملفات الخاصّة بالقاعدة، بعد الانتهاء قم بإعادة إقلاع القاعدة من جديد.

النسخ الاحتياطي لملفّات المعطيات في النمط المؤرشف ARCHIVELOG

يمكن إجراء نسخ احتياطي لكامل القاعدة بنفس الطريقة التي ذكرناها في الفقرة السابقة. كذلك يمكن إجراء النسخ الاحتياطي لجزء من القاعدة كالنسخ الاحتياطي لفضاء من الفضاءات الجدولية عندما تكون فعّالة أو غير فعّالة.

١. **النسخ الاحتياطي لفضاء جدولي غير فعّال Offline Tablespace:**

☆ حدّد أولاً أسماء الملفات الخاصة بالفضاء الجدولي X على الشكل:

```
SELECT file_name
FROM sys.dba_data_files
WHERE tablespace_name='X';
```

☆ قم بعد ذلك بإلغاء تأهيل الفضاء الجدولي X:

```
ALTER TABLESPACE X OFFLINE NORMAL;
```

☆ استخدم أحد برامج النسخ الاحتياطي لإنشاء نسخة احتياطية من ملفات هذا الفضاء الجدولي على قرص التخزين.

☆ قم أخيراً بإعادة تأهيل الفضاء الجدولي X:

```
ALTER TABLESPACE X ONLINE;
```

٢. النسخ الاحتياطي لفضاء جدولي فعّال Online Tablespace:

☆ حدّد أولاً أسماء الملفات الخاصة بالفضاء الجدولي X على الشكل:

```
SELECT file_name
FROM sys.dba_data_files
WHERE tablespace_name='X';
```

☆ قم بعد ذلك بإعلام أوراكل عن بدء عملية النسخ الاحتياطي للفضاء الجدولي X بكتابة التعليمة:

```
ALTER TABLESPACE X BEGIN BACKUP;
```

☆ استخدم أحد برامج النسخ الاحتياطي لإنشاء نسخة احتياطية من ملفات هذا الفضاء الجدولي على قرص التخزين.

☆ قم أخيراً بإعلام أوراكل عن انتهاء عملية النسخ الاحتياطي للفضاء الجدولي X بكتابة التعليمة:

```
ALTER TABLESPACE X END BACKUP;
```

النسخ الاحتياطي لملفات التحكّم

يمكن بسهولة إجراء النسخ الاحتياطي لملفّات التحكّم عندما تكون قاعدة المعطيات مغلقة وذلك باستخدام أيّ من برامج النسخ الاحتياطي المتوفرة لديك.

أما في حال الحاجة لإجراء النسخ الاحتياطي لملفات التحكّم والقاعدة مفتوحة فيمكنك استخدام التعليمة:

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

والتي تقوم بتوليد أوامر توليد ملفات التحكّم,

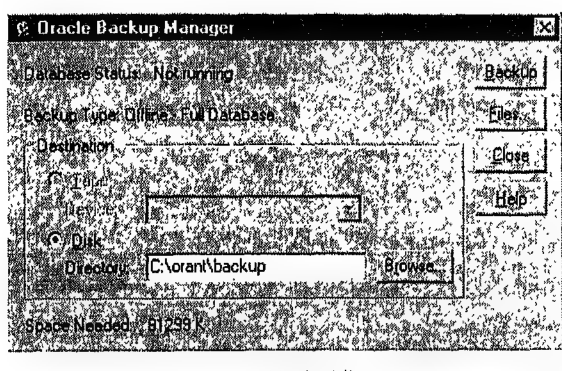
يمكنك أيضاً استخدام التعليمة:

ALTER DATABASE
BACKUP CONTROLFILE controlfile_name [REUSE];

لتوليد نسخة من ملف التحكم controlfile_name.

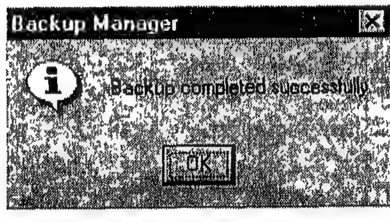
النسخ الاحتياطي باستخدام الأداة Oracle Backup Manager

يمكن استخدام الأداة Oracle NT Backup الموجودة ضمن مجموعة برامج Oracle for NT، وذلك من خلال المخدّم لإجراء النسخ الاحتياطي لقاعدة المعطيات بشكل بسيط وسهل.



الشكل ٤٣-١

قم أولاً بتشغيل هذا البرنامج مع تسجيل الدخول بالحساب INTERNAL. فإذا كنت تعمل بنمط NOARCHIVELOG تظهر نافذة تشبه الشكل ٤٣-١، تمكّنك هذه النافذة من إجراء نسخ احتياطي لكامل القاعدة Full Offline Backup فقط. أما عندما تكون القاعدة بنمط ARCHIVELOG، فتستطيع إجراء إما نسخ احتياطي لكامل القاعدة Offline-Full Database، أو نسخ احتياطي لأحد الفضاءات الجدولية الفعّال Online Tablespace، أو نسخ احتياطي لملفات التحكّم الفعّال Online-Control File Only.



الشكل ٤٣-٢

عند الانتهاء من عملية النسخ الاحتياطي تظهر رسالة تشبه الشكل توضح فيما إذا كانت عملية النسخ قد تمّت بشكل سليم أم لا (انظر الشكل ٤٣-٢).

النسخ الاحتياطي باستخدام الأداة Backup Manager

يمكـــن استخدم أحد أهم أدوات Enterprise Manger وهي الأداة Backup Manager.

يمكن استخدام هذه الأداة إما من الحاسوب المخدّم أو الزبون.

توجد مجموعة من الخطوات التي يجب القيام بها قبل البدء بعملية النسخ الاحتياطي.

١. إنشاء مجلّد الاسترداد recovery catalog.

يجب أولاً إنشاء مجلّد الاسترداد باتباع الخطوات:

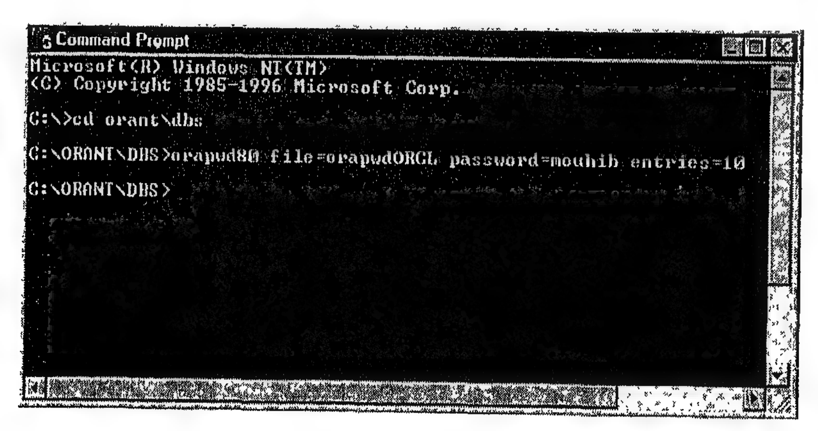
```
SPOOL recovery.log
CREATE USER rman IDENTIFIED BY rman
TEMPORARY TABLESPACE temporary
DEFAULT  TABLESPACE recovery_cat
        QUOTA unlimited ON recovery_cat;
GRANT recovery_catalog_owner TO rman;
CONNECT rman/rman
@d:\orant\rdbms80\catrman
```

٢. توليد ملف جديد لكلمات المرور password file:

قم أولاً بالانتقال إلى المجلد orant\dbs، احفظ ملف كلمات المرور القديم باسم جديد (يأخذ عادةً الاسم orapwSID)، ثمّ قم بإنشاء ملف كلمات مرور جديد باستخدام التعليمة:

```
ORAPWD80 file=orapwSID password=password entries=10
```

انظر الشكل ٤٣-٣:



الشكل ٤٣-٣

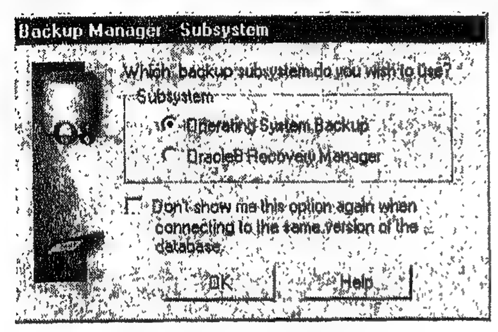
غيّر أو أضف السطر التالي إلى ملف وسطاء التمهيد initSID.ora:

```
remote_login_passwordfile = exclusive
```

أنشئ بعد ذلك المستخدم البعيد وأعطه سماحيات مدير القاعدة:

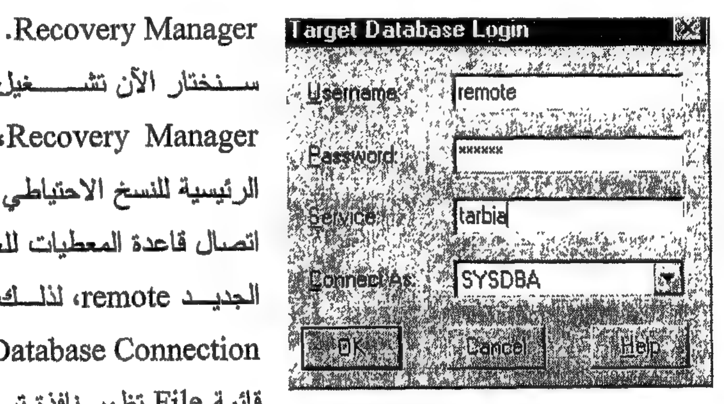
GRANT connect, resource TO remote;
GRANT dba TO remote;

الآن بعد أن قمنا بإنشاء المستخدم remote، يجب جعل Backup Manager يقوم باستخدام هذا الحساب. من أجل ذلك شغّل Backup Manger من Enterprise manager، تظهر نافذة تشبه الشكل ٣٤-٤.



الشكل ٤٣-٤

نستطيع من خلال هذه النافذة استخدام أداة النسخ الاحتياطي الخاصة بنظام التشغيل Operating System Backup، أو مدير الاسترداد ضمن أوراكل٨ Oracle8 Recovery Manager.



الشكل ٤٣-٥

سنختار الآن تشغيل الأداة Oracle8 Recovery Manager، تظهر النافذة الرئيسية للنسخ الاحتياطي. نقوم أولاً بتغيير اتصال قاعدة المعطيات للعمل مع المستخدم الجديد remote، لذلك اطلب الأمر Change Database Connection من قائمة File تظهر نافذة تسجيل المستخدم الجديد كما في الشكل ٤٣-٥.

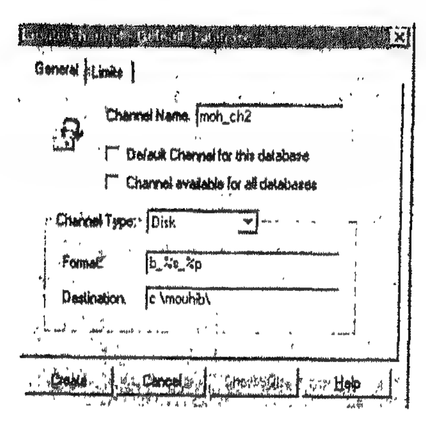
يجب أيضاً الاتصال بمجلّد الاسترداد الذي تمّ إنشاؤه. من أجل ذلك اطلب الخيار Catalog Connect String من قائمة Catalog. تظهر نافذة تسجيل مجلّد الاسترداد والتي تشبه إلى حدّ كبير النافذة السابقة، حدّد اسم المستخدم وكلمة المرور واسم الخدمة المتعلقة بهذا المجلّد.

يجب ملاحظة أنّه في التطبيقات الكبيرة، يجب أن يكون مجلّد الاسترجاع في قاعدة معطيات مختلفة عن القاعدة الحالية.

بعد إجراء الاتصال، يجب تسجيل مجلّد الاسترداد وذلك بطلب الأمر Register من قائمة Catalog. الآن أصبحت جاهزاً لاستخدام Oracle Backup Manager.

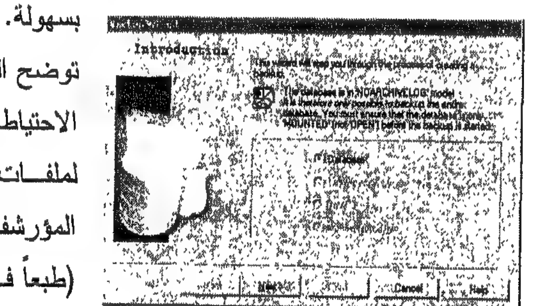
تشغيل برنامج Backup Manger

عندما تقوم بتشغيل البرنامج لأول مرة، تظهر رسالة توضح بأنّك لم تقم بتعريف قنوات Channels من أجل عمليات الإدخال والإخراج. قم بتعريف قناة جديدة وذلك بتحديد اسم القناة Channel Name، ونمطها Channel Type، وتنسيقها Format ومجلد الهدف Destination كما في الشكل ٤٣-٦.



الشكل ٤٣-٦

بعد إنشاء قناة واحدة على الأقل، أصبح بإمكانك استخدام معالج النسخ الاحتياطي بسهولة.

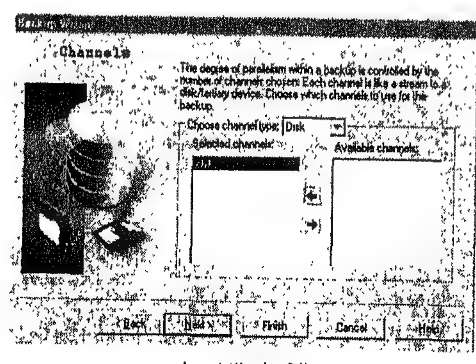


الشكل ٤٣-٧

توضح النافذة الأولى هل سيتم إجراء النسخ الاحتياطي لكامل القاعدة أو لفضاء جدولي أو لملفات المعطيات أو لملفات الإرجاع المؤرشفة، انظر الشكل ٤٣-٧.

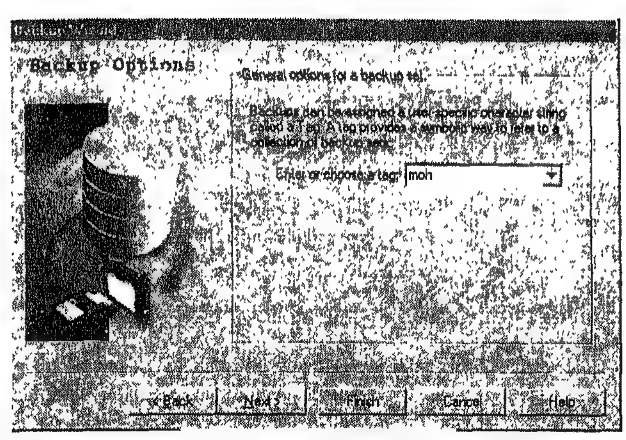
(طبعاً في حال تشغيل القاعدة بنمط NOARCHIVELOG فلن يكون سوى الخيار الأول فعّال).

أما النافذة الثانية فتمكننا من تحديد قناة الاتصال (انظر الشكل ٤٣-٨).



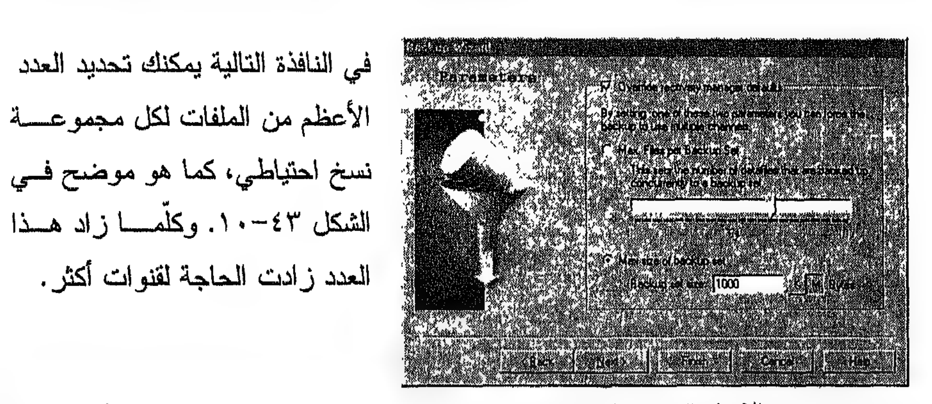
الشكل ٤٣-٨

أما في النافذة التالية فيجب تحديد معرف tag لمجموعة النسخ الاحتياطي، وعندما تقوم بإجراء نسخ احتياطي متزايد، يمكنك هنا تحديد مستوى النسخ الاحتياطي (انظر الشكل ٤٣-٩).



الشكل ٤٣-٩

في النافذة التالية يمكنك تحديد العدد الأعظم من الملفات لكل مجموعة نسخ احتياطي، كما هو موضح في الشكل ٤٣-١٠. وكلّما زاد هذا العدد زادت الحاجة لقنوات أكثر.

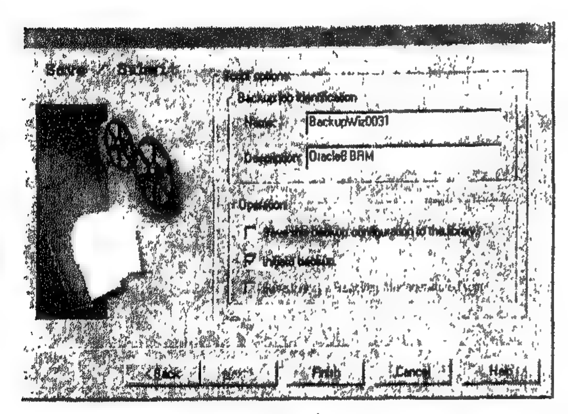


الشكل ٤٣-١٠

بعد ذلك يمكنك جدولة عملية النسخ الاحتياطي كما توضح النافذة ٤٣-١١.

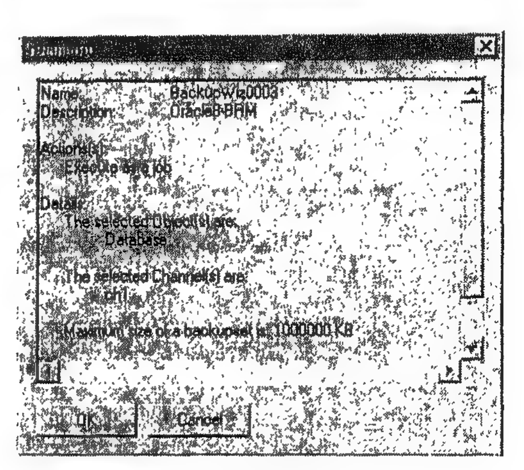
الشكل ٤٣-١١

النافذة الأخيرة في المعالج تساعد على حفظ توصيف النسخ الاحتياطي (انظر الشكل ٤٣-١٢).



الشكل ٤٣-١٢

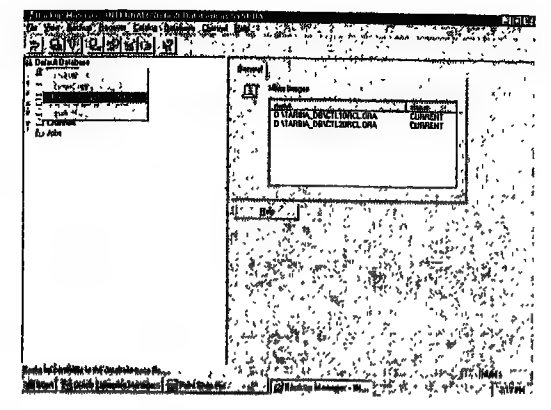
عند النقر على زر Finish تظهر نافذة تلخيصيّة توضح الخطوات التي تمّ تحديدها للقيام بالنسخ الاحتياطي (انظر الشكل ٤٣-١٣).




الشكل ٤٣-١٣



النسخ الاحتياطي لملفات التحكّم



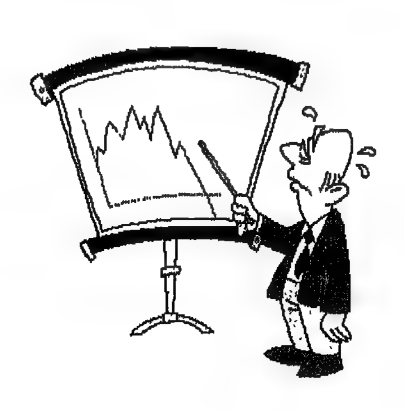
الشكل ٤٣-١٤

يمكن من النافذة الرئيسية لمدير النسخ الاحتياطي Backup manager إجراء عملية النسخ الاحتياطي لملفات التحكم.
انقر بزر الفأرة الأيمن على ملفات التحكم واختر الأمر Backup Controlfile to Trace، أو اختر الأمر السابق من قائمة Backup كما في الشكل ٤٣-١٤.

النسخ الاحتياطي وقاموس المعطيات

يحتوي قاموس المعطيات على:

☆ المشهد المتعلقة بالنسخ الاحتياطي: V$BACKUP.







الاسترداد
Recovery

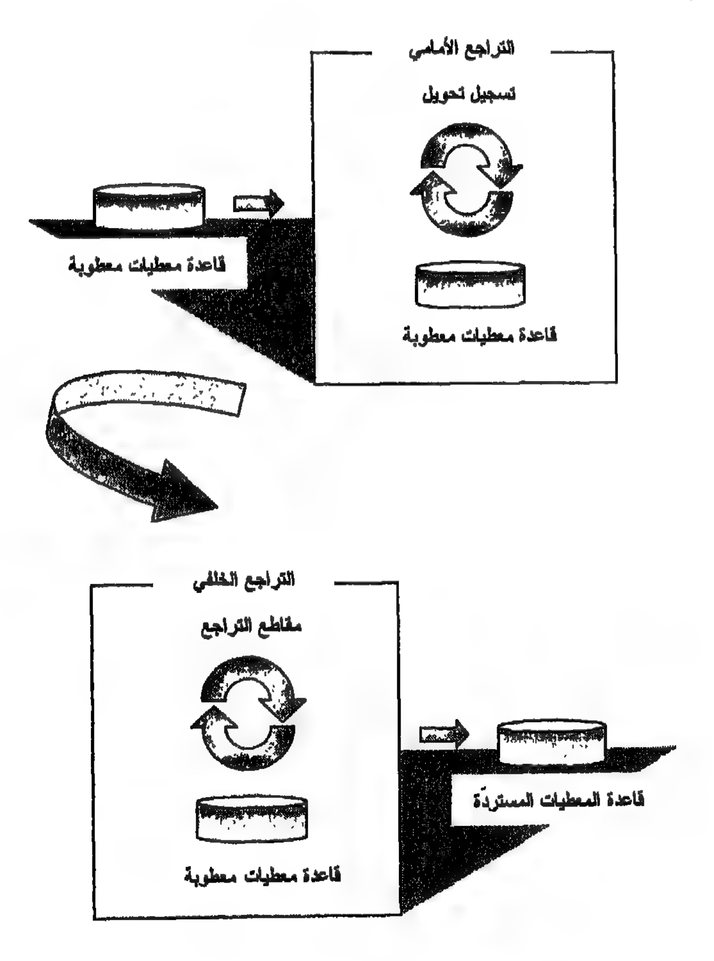
يمكنك في حال حدوث أي خطأ في قاعدة معطياتك، إجراء العمليات اللازمة لاسترداد هذه القاعدة. وتوجد أنماط عديدة لعلمية الاسترداد، كما تتضمن عملية الاسترداد اتباع مرحلتين: التراجع الأمامي Roll-Forward، والتراجع الخلفي Roll-Back.

ففي مرحلة التراجع الأمامي يقوم مدير الاسترداد Recovery Manager بتطبيق العمليات الموجودة في مجموعات تسجيل التحويلات من أجل إعادة تنفيذ التحويلات المثبتة غير الموجودة في ملفات المعطيات الحالية لقاعدة المعطيات.

أمّا في حال حدوث تعطل بسيط في القاعدة فقد تكفي عملية إرجاع التحويلات لاسترجاع ماتمّ فقدانه من القاعدة، وعند حدوث أعطال خطيرة فقد تحتاج لاستخدام النسخة الاحتياطية من ملفات المعطيات ومجموعات التسجيل المؤرشفة من أجل إتمام مرحلة التراجع الأمامي.

أما في مرحلة التراجع الخلفي Roll-Back يقوم أوراكل باستخدام المعلومات الموجودة في مقاطع التراجع في أجل التراجع عن التغييرات التي تمّت على القاعدة من قبل أي تحويل لم يتمّ تثبيته عند حدوث العطل. في هذه المرحلة تعود القاعدة إلى الحالة التي كانت عليها عند آخر عملية تثبيت تمّ إجراءها على التحويلات قبل حدوث المشكلة.

يوضح الشكل ٤٤-١ كيفية عمل مرحلتي التراجع الأمامي والتراجع الخلفي من أجل إنجاز عملية استرداد القاعدة.



شكل ٤٤-١

توجد أنماط عديدة للاسترداد أهمها:

- **الاسترداد المعطوب Crash Recovery**: وهي عملية الاسترداد التلقائية التي يقوم أوراكل باستخدامها للاسترداد من تعطلٍ بسيط للقاعدة، كمشاكل تعطّل الكهرباء ومشاكل البرمجيات والتي تحدث بشكل متكرر.

في هذه الحالة لايقوم مخدّم قاعدة المعطيات بإطفاء القاعدة بشكل نظامي. فلربما يقوم بإدارة العديد من التحويلات المفتوحة التي تقوم بإجراء التعديلات على معلومات القاعدة. إضافةً إلى ذلك فقد لايكون هناك متسع من الوقت كي يقوم أوراكل بإنجاز نقاط التدقيق على قاعدة المعطيات للتحقّق من أن جميع كتل المعطيات التي تمّ تعديلها في الذواكر المؤقتة قد ثبتت بأمان على ملفات المعطيات. لذلك قد تصبح المعطيات الموجودة في ملفات المعطيات غير متجانسة.

هنا يقوم أوراكل بإنجاز مرحلتي التراجع الأمامي والتراجع الخلفي لإتمام عملية استرداد القاعدة وتتم العملية بشفافية تامّة بدون أي تدخل من قبل مدير القاعدة.

- **استرداد الوسائط Media Recovery**: قد يتم ضياع ملف أو أكثر من ملفات المعطيات إما بسبب تعطل القرص أو بسبب خطأ المستخدم، في هذه الحالة فأنت بحاجة لإجراء عملية استرداد الوسائط والتي تتضمن الخطوات التالية:

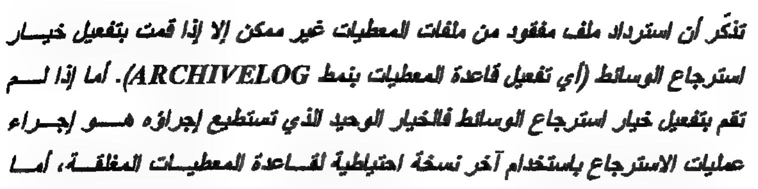
١. إصلاح جميع المشاكل في التجهيزات.

٢. استرجاع ملفات المعطيات الضائعة: حيث يمكّنك استخدام مدير الاسترداد Recovery Manager ومجموعة النسخة الاحتياطية أو النسخ المطابقة Image Copies لاسترجاع ملفات المعطيات الضائعة.

٣. بعد أن تقوم باسترجاع ملفات المعطيات الضائعة، يجب على أوراكل استرداد الملفات بإنجاز مرحلتي الاسترداد: الامتداد الأمامي والامتداد الخلفي.

وبالاعتماد على الفترة التي انقضت على آخر عملية نسخ احتياطي وكمية العمل التي تمّ إنجازها بعد ذلك، فقد تحتاج عملية الاسترداد إلى عدد كبير من مداخل الإرجاع. لذلك يجب عليك التأكد من أن جميع مجموعات التسجيل المؤرشفة متاحة لمدير الاسترداد Recovery Manager والذي يقوم باستخدام المعلومات المتوفرة في مجلد الاسترداد لاسترجاع ملفات التسجيل المؤرشفة من مجموعات النسخ الاحتياطي قبل بدء عملية الاسترداد.

٤. إنجاز العمليات الضرورية لاسترداد الوسائط.


تذكّر أن استرداد ملف مفقود من ملفات المعطيات غير ممكن إلا إذا قمت بتفعيل خيار استرجاع الوسائط (أي تفعيل قاعدة المعطيات بنمط ARCHIVELOG). أما إذا لم تقم بتفعيل خيار استرجاع الوسائط فالخيار الوحيد الذي تستطيع إجراؤه هو إجراء عمليات الاسترجاع باستخدام آخر نسخة احتياطية لقاعدة المعطيات المغلقة، أما التحويلات المثبتة التي تم إجراؤها بعد آخر عملية نسخ احتياطي فستضيع للأبد.

الاسترداد الكامل Complete Recovery

وهي عملية استرداد عمل جميع التحويلات المثبتة، وتتضمن استرداد قاعدة المعطيات database recovery واسترداد الفضاء الجدولي tablespace recovery واسترداد ملف المعطيات data file recovery.

* **استرداد قاعدة المعطيات:** وهي أبسط عملية لاسترداد كافة الأعمال الضائعة في قاعدة المعطيات بخطوة واحدة فقط وذلك باستخدام مدير الاسترداد Recovery Manager الذي يقوم بتحديد ملفات المعطيات المعطوبة واستردادها تلقائياً باستخدام المعلومات الموجودة في مجلد الاسترداد ومجموعات النسخ الاحتياطي لملفات المعطيات إضافةً إلى مجموعات تسجيل التحويلات.

يجب الانتباه إلى أنه لإجراء عملية استرداد كامل يجب أن تكون قاعدة المعطيات مركبة Mounted لكن مغلقة Closed.

* **استرداد الفضاءات الجدولية Tablespace Recovery:** في حال تعطّل جزء من قاعدة المعطيات، يمكنك استرداد العمل الضائع في جميع ملفات المعطيات المتعلقة بفضاء جدولي محدد باستخدام نمط استرداد الفضاء الجدولي. ويمكنك القيام بذلك عندما تكون قاعدة المعطيات مفتوحة والفضاء الجدولي المعطوب في حالة offline، أو عندما تكون قاعدة المعطيات مركبة لكن مغلقة.



في حال حدوث مشكلة في أي ملف معطيات في الفضاء الجدولي SYSTEM فلن تعمل قاعدة المعطيات بشكل سليم، لذلك يجب عليك إطفاء قاعدة المعطيات وإجراء عملية الاسترداد والقاعدة مغلقة.

* **استرداد ملفات المعطيات DataFile Recovery**: تستطيع استخدام مدير الاسترداد Recovery Manager لإجراء عملية استرداد ملف معطيات معطوب وذلك عندما تكون قاعدة المعطيات مفتوحة والفضاء الجدولي المرتبط بهذا الملف في حالة offline أو عندما تكون قاعدة المعطيات مركبة لكن مغلقة.

الاسترداد غير الكامل Incomplete Recovery

في ظروف معينة قد تحتاج لاسترداد جزء من العمل المتعلق بالتحويلات المثبتة وذلك بتطبيق كمية محددة من مداخل الإرجاع redo entries في مجموعة تسجيل تحويلات قاعدة المعطيات.

توجد ثلاثة أنماط من الاسترداد غير الكامل:

* **الاسترداد الوقتي Time_Based Recovery**: ويسمى أيضاً point_in_time recovery والذي يقوم باسترداد عمل التحويلات المثبتة في قاعدة المعطيات اعتباراً من وقت محدد، مثلاً منذ يوم الأحد الساعة 8:05 قبل الظهر قبل أن يقوم المستخدم بحذف جدول هام.

* **الاسترداد المتغير Change_ Based Recovery**: ويتم من خلال هذا النمط استرداد عمل التحويلات المثبتة في قاعدة المعطيات اعتباراً من رقم تغيير نظام محدد SCN (System Change Number).

فكما نعرف فإن أوراكل يحدد لكل تحويل يقوم بتثبيته برقم SCN وحيد. فإذا عرفت رقم SCN الموافق للتحويل الضائع يمكنك عندها استخدام نمط الاسترداد هذا لاسترداد التحويل الضائع.

* **الاسترداد الملغى Cancel_ Based Recovery**: يتم من خلال هذا النمط استرداد عمل التحويلات المثبتة في قاعدة المعطيات اعتباراً من تطبيق مجموعة تسجيل محددة. لذلك يجب تحديد تسلسل التسجيل الأخير last log sequence وتطبيق هذا النمط من الاسترداد.

الاسترداد باستخدام الأداة Server Manager

الاسترداد في النمط غير المؤرشف NOARCHIVELOG

في حال ضياع أحد ملفات المعطيات، فيمكنك استخدام النسخة الاحتياطية لقاعدة المعطيات المغلقة. أما في حال ضياع أحد ملفات الإرجاع فيمكنك استرجاعه باستخدام أحد النسخ المطابقة أو إجراء استرداد لكامل القاعدة.

أخيراً في حال ضياع أحد ملفات التحكّم فيمكنك استخدام النسخ المضاعفة، أو إعادة توليد ملف التحكّم أو استرداد كامل قاعدة المعطيات.

الاسترداد في النمط المؤرشف ARCHIVELOG

يمكن إجراء عملية الاسترداد باستخدام تعليمة ALTER DATABASE مع الخيار RECOVER على الشكل التالي:

```
[ALTER DATABASE]
        RECOVER [AUTOMATIC] [FROM 'location']
        {[DATABASE]          [UNTIL CANCEL
                             |UNTIL TIME date
                             |UNTIL CHANGE integer
                             |UNTIL BACKUP CONTROLFILE]
        |TABLESPACE tablespace[,tablespace]...
        |DATAFILE 'filename'[,'filename']...
        |CONTINUE [DEFAULT]}
```

حيث:

- ☆ **AUTOMATIC**: لإدارة ملفات الإرجاع التي سيتم تطبيقها تلقائياً.
- ☆ **FROM location**: لتحديد موقع الأرشيف.
- ☆ **DATABASE**: لاسترداد كامل قاعدة المعطيات (الخيار الافتراضي).
- ☆ **UNTIL CANCEL**: لإجراء عملية الاسترداد اعتباراً من آخر ملف إرجاع.
- ☆ **UNTIL TIME date**: للاسترداد حتى وقت حدوث العطل.
- ☆ **UNTIL CHANGE integer**: للاسترداد حتى آخر تحويلات قبل SCN.

☆ UNTIL BACKUP CONTROLFILE: لاستخدام النسخة المحفوظة من ملف التحكّم بدلاً من النسخة الحالية.

☆ TABLESPACE tablespace: لاسترداد الفضاء الجدولي tablespace فقط.

☆ DATAFILE filename: لاسترداد ملف معطيات.

☆ LOGFILE filename: لمتابعة الاسترداد مع ملف الإرجاع filename.

☆ CONTINUE [DEFAULT]: لمتابعة تطبيق الإرجاع تلقائياً.

❋ مثلاً في حال حدوث عطل في قرص التخزين أدى لتوقف العمل مع قاعدة المعطيات، فيمكننا إجراء استرداد لكامل قاعدة المعطيات المغلقة باتباع الخطوات التالية:

١. إيقاف قاعدة المعطيات.

٢. إصلاح العطل في قرص التخزين.

٣. استرجاع آخر نسخة احتياطية وملفات الأرشيف.

٤. إعادة إقلاع القاعدة بحالة MOUNT.

٥. إعادة تسمية أو تغيير مواقع الملفات باستخدام التعليمة:

ALTER DATABASE RENAME FILE ... TO ...

٦. بدء عملية الاسترداد :

```
RECOVER     AUTOMATIC
            DATABASE ...;
```

٧. إعادة فتح القاعدة بالحالة العادية.

ALTER DATABASE OPEN;

❋ أما في حال حدوث عطب ما أدى لتلف أحد ملفات الفضاءات الجدولية، بينما بقيت ملفات الفضاء الجدولي SYSTEM سليمة، كذلك الملفات التي تحوي مقاطع التراجع. فيمكن إجراء استرداد لكامل قاعدة المعطيات المفتوحة مع إلغاء تأهيل الفضاء الجدولي المعطوب باتباع الخطوات التالية:

١. في حال لم تكن القاعدة مفتوحة، قم بإعادة فتحها واجعل الفضاء الجدولي المتعلق بالملفات المعطوبة غير فعّال.

```
connect internal
STARTUP MOUNT ...
ALTER DATABASE OPEN;
ALTER TABLESPACE tablespace OFFLINE;
```

٢. قم بإصلاح العطل.

٣. استبدل الملفات المعطوبة بنسخها الاحتياطية.

٤. قم باسترداد الملفات المعطوبة.

```
RECOVER    TABLESPACE
           tablespace1,..., tablespace n;
RECOVER    DATAFILE datafile1,..., datafile n;
```

٥. قم بإعادة تأهيل الفضاء الجدولي.

```
ALTER DATABASE tablespace1,...,tablespace n ONLINE;
```

* كذلك فإنّه في حال ضياع ملفات الإرجاع، أو الحذف غير المقصود لجدول، يمكن إجراء عملية استرداد غير كاملة باتباع الخطوات التالية:

١. إغلاق القاعدة مع الخيار ABORT.

```
connect internal
shutdown abort
```

٢. حفظ جمع ملفات القاعدة.

٣. إعادة إقلاع الهيئة (بحالة MOUNT فقط).

٤. إعادة تسمية وتغيير موقع الملفات المعطوبة.

٥. تفعيل الملفات المعطوبة ONLINE، في حال القيام بإجراء استرداد اعتماداً على الزمن أو SCN.

٦. تنفيذ التعليمات التالية لإجراء الاسترداد:

```
RECOVER database UNTIL CANCEL;
```

أو

```
RECOVER database
        UNTIL TIME 'YYY-MM-HH24:MI:SS';
```

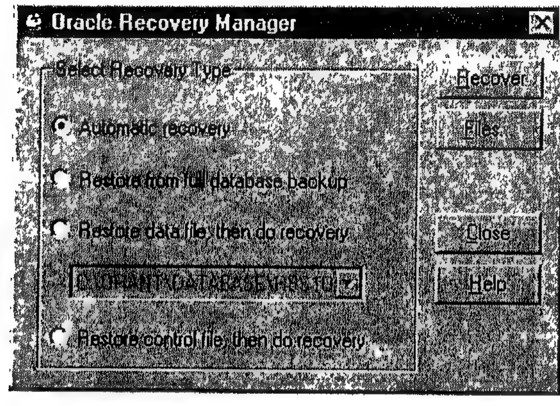
أو

```
RECOVER database
```

UNTIL CHANGE SCN;

الاســـترداد باســــــتخدام الأداة Oracle Recovery Manager

يمكن استخدام الأداة Oracle NT Recovery الموجودة ضمن مجموعة برامـج Oracle for NT، وذلك من خلال المخدّم لإجراء الاسترداد لقاعدة المعطيات بشكل بسيط وسهل.



الشكل ٤٤-١

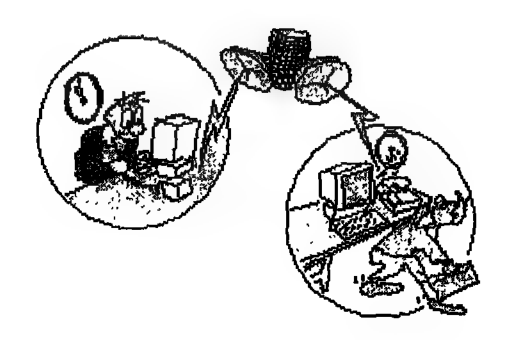
قم أولاً بتشغيل هذا البرنـامج مع تسجيل الدخول بالحسـاب INTERNAL. تظهر نافذة تشبه الشكل ٤٤-١، تستطيع من خلال هذه النافذة إجـراء استرداد تلقائي Automatic recovery، أو الاســترجاع من نسخة احتياطيــة كاملــة لقاعدة المعطيـات Restore from full database backup، أو استرجاع ملف معطيات ومن ثم إجــراء الاســترداد Restore data file, then do recovery، أو استرجاع ملف تحكّم ومـــن ثـمّ إجــراء الاسترداد Restore control file, then do recovery.

عند الانتهاء من عملية الاسترداد تظهر رسالة توضح فيما إذا كانت عملية الاســترداد قــد تمّت بشكل سليم أم لا.

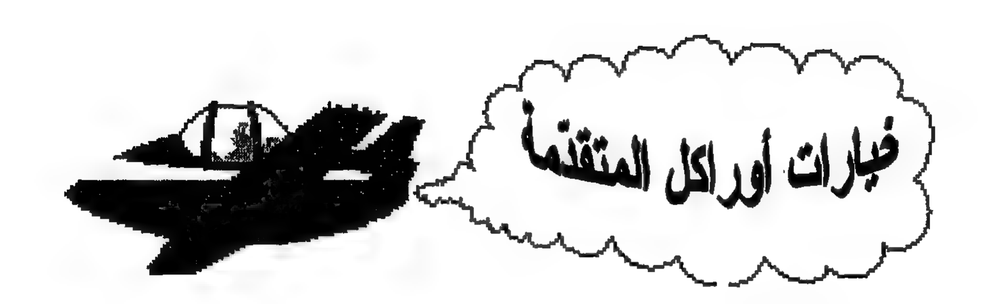
الاسترداد وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة للاسترداد هي:

☆ المشاهد المتعلقة بالاسترداد: V$RECOVERY_FILE.




الباب الثاني عشر



٤٥. أوراكل٨ وقواعد المعطيات الموزّعة.

٤٦. أوراكل٨ وتناسخ المعطيات.

٤٧. أوراكل٨ وقواعد المعطيات المتوازية.



أوراكل٨ وقواعد المعطيات الموزّعة

Oracle8 and Distributed Data Bases

مع زيـــادة حجــم الشبكات الحاسوبية وارتباطها بعدد هائل من الحواسيب الشخصية. أصبح العديد من الشركات (لا سيما الكبيرة منها) بحاجة إلى تخزين معلوماتها في عـــدة قواعد معطيات بدلاً من تركيزها في قاعدة معطيات وحيدة، بالتالي ظهرت هناك حاجة لتقسيم المعلومات الموجودة في قاعدة معطيات واحدة إلى عدة قواعد معطيات موجودة في عدة مواقع فيزيائية مختلفة.

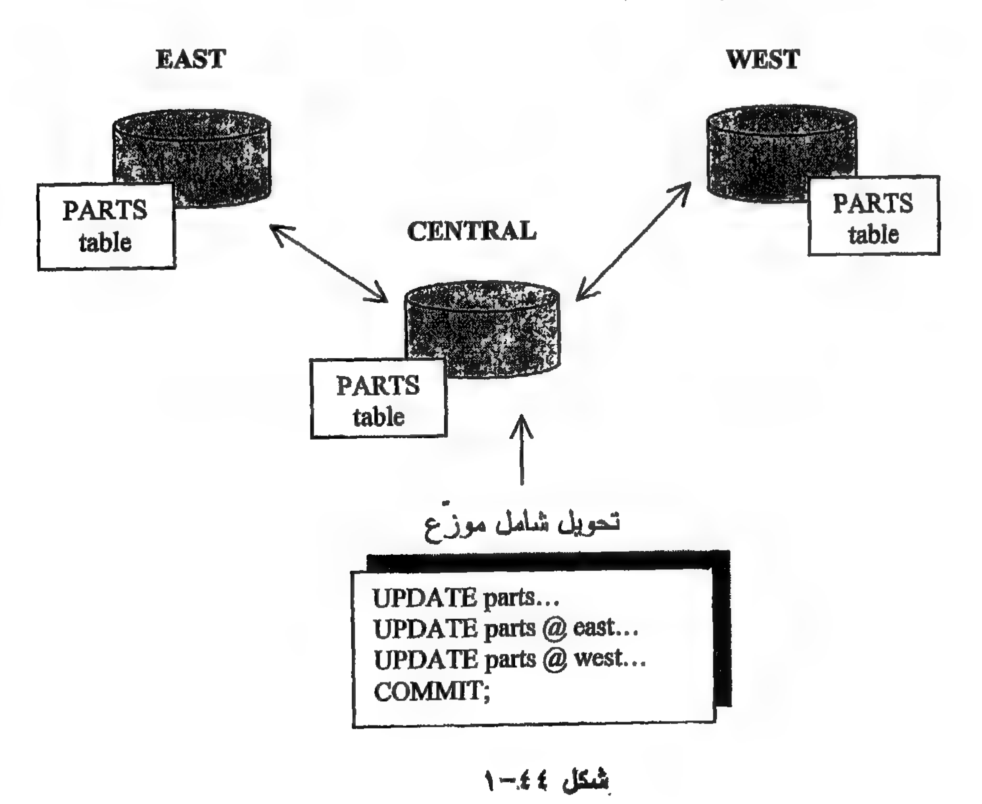
الآن حـــتى تســـتطيع التعامل مع جميع قواعد المعطيات هذه وكأنها قاعدة معطيات واحدة وإتاحـــة المعـــلومات الموجـــودة فيهـــا لجميع مستخدمي النظام فإنّه يمكنك استخدام قاعدة معطيـــات أوراكل الموزّعة وميزة تناسخ المعطيات Data replication. سنقوم في هذا الفصـــل بشـــرح بنية نظام قاعدة معطيات أوراكل الموزّعة وكيف يمكنك استخدام تناسخ المعطيات لتحديد وتحديث نسخ المعطيات المحلية في قواعد المعطيات المتعددة.

بنيــة قــاعدة معطيــات أوراكــل الموزّعـــة Oracle Distributed Database Architecture

قاعدة المعطيات الموزّعة عبارة عن مجموعة من قواعد المعطيات التي تبدو للمســتخدمين والتطبيقات وكأنها قاعدة معطيات وحيدة.

في أغلب الحالات، فإن قواعد المعطيات الموزّعة تتوضّع في حواسيب منفصلة تتصل فيما بينها عن طريق شبكة.

بعد أن تقوم بتوصيف نظام قاعدة معطيات أوراكل الموزّعة، تصبــح جميــع المعطيــات الموجودة في النظام متاحة للتطبيقات وكأنها متوضّعة في قاعدة معطيات منطقية واحــــدة, فمثلا يمكن للتحويل الشامل الموجود في الشكل ٤٤-١ أن يحتوي تعليمــات DML تقــوم بتعديل معطيات عدة قواعد معطيات.

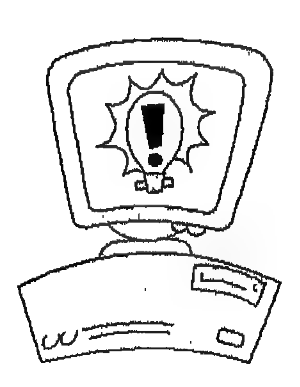


شكل ٤٤-١

يقوم كل مخدّم قاعدة معطيات في نظام قاعدة معطيات موزّعة بإدارة الوصول إلى قاعدة معطياته المحلية ولا يمكن لأي مخدّم من المخدّمات إدارة كامل النظام، كما يجب على جميع المخدّمات في النظام التعاون فيما بينها لتحقيق التناسق والدقة.

يمكننا إذا اعتبار أنظمة قواعد معطيات أوراكل الموزّعة كتوسيع لنموذج المخدّم/الزبون Client/Server Model لأن مخدّم قاعدة المعطيات في نظام قواعد معطيات موزّعة يستطيع العمل كزبون.

فمثلا إذا تعمقنا في الشكل ٤٤-١، فإننا نجد أنه عندما يقوم الحاسوب الذي يدير قاعدة المعطيات CENTRAL بتنفيذ التعليمة الأولى في التحويل ، فإنه يعمل كمخدّم قاعدة معطيات لأن هذه التعليمة تستطيع الوصول إلى جدول PARTS في قاعدة المعطيات المحلية. أما عندما يقوم بتنفيذ التعليمتين الثانية والثالثة فإن نفس الحاسوب يعمل الآن كزبون لأنه يقوم بتوجيه تعليمات SQL إلى مخدّمات قاعدتي المعطيات WEST, EAST لإجراء التعديل عن بعد لجداول PARTS. وتقوم جميع المخدّمات في نظام قاعدة معطيات أوراكل الموزّعة باستخدام برمجيات تشبيك أوراكل Net8 حتى تستطيع الاتصال فيما بينها خلال الشبكة.



خدمات قاعدة المعطيات والتسمية في قاعدة معطيات موزّعة

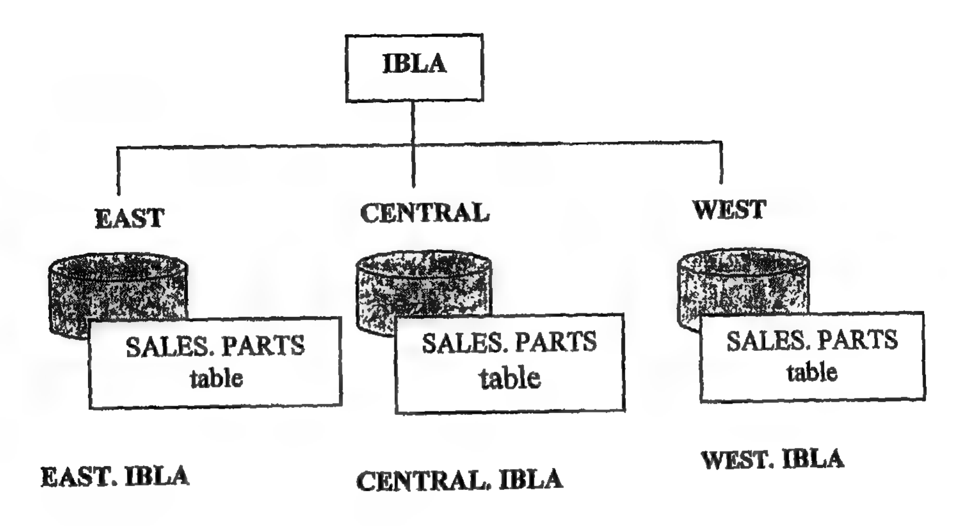
يجب أن تمتلك جميع الخدمات (صفوف انتظار الطباعة، مخدمات البريد وغيرها) المتاحة على الشبكة على أسماء وحيدة حتى تتمكن التطبيقات وكذلك المستخدمين من معرفة كيفية الوصول إليها.

في نظام قاعدة المعطيات الموزّعة، فإن مخدّم قاعدة المعطيات أو الممثل هو ببساطة عبارة عن خدمة قاعدة معطيات database service متاحة على الشبكة، لذلك يجب أن يكون اسم هذه الخدمة وحيداً حتى نستطيع استخدامها. هذا الاسم ليس سوى اسم قاعدة المعطيات العام global database name والمؤلف من جزأين:

☆ **الجزء الأول:** هو اسم قاعدة المعطيات الأساسي الذي يعطى لها عند إنشائها، ولا يمكن لهذا الجزء أن يتجاوز ثمانية أحرف.

☆ **الجزء الثاني:** هو مجال شبكة قاعدة المعطيات database network domain والذي يحدد الموقع المنطقي لقاعدة المعطيات ضمن الشبكة.

يوضح الشكل ٤٤-٢، شبكة قواعد معطيات في شركة افتراضية اسمها IBLA، تتضمن



شكل ٤٤-٢

هذه الشبكة ثلاث قواعد معطيات WEST, CENTRAL , EAST.

أسماء قاعدة المعطيات العامة الموافقة (لأسماء الخدمات) هي: WEST.IBLA, CENTRAL.IBLA, EAST.IBLA.

حتى يمكنك تحديد عناصر مخطط محددة في قاعدة معطيات موزّعة ليست محليّة ضمن قاعدة المعطيات الخاصة بك، يجب عليك توسيع اسم العنصر باستخدام اسم قاعدة المعطيات العام المتعلّق به.

مثلاً كما ترى في الشكل ٢-٤٤ هنالك جدول PARTS في كل من قواعد المعطيات WEST, CENTRAL, EAST، فإذا كنت تقوم بتشغيل تطبيق (كتطبيق SQL * PLUS مثلاً) وقمت بالاتصال بقاعدة معطيات CENTRAL، يمكنك استعلام جدول PARTS في قاعدة المعطيات EAST بتحديد اسم العنصر بشكل كامل ضمن قاعدة المعطيات الموزّعة، مثلاً:

*SELECT * FROM sales.Parts @ east.ibla*

الآن حتى يقوم النظام بتنفيذ الاستعلام، فإن مخدّم قاعدة معطيات CENTRAL المحلي يقوم بشكل ضمني باستخدام ارتباط قاعدة المعطيات database link الذي يقوم بإجراء الاتصال مع قاعدة المعطيات EAST.

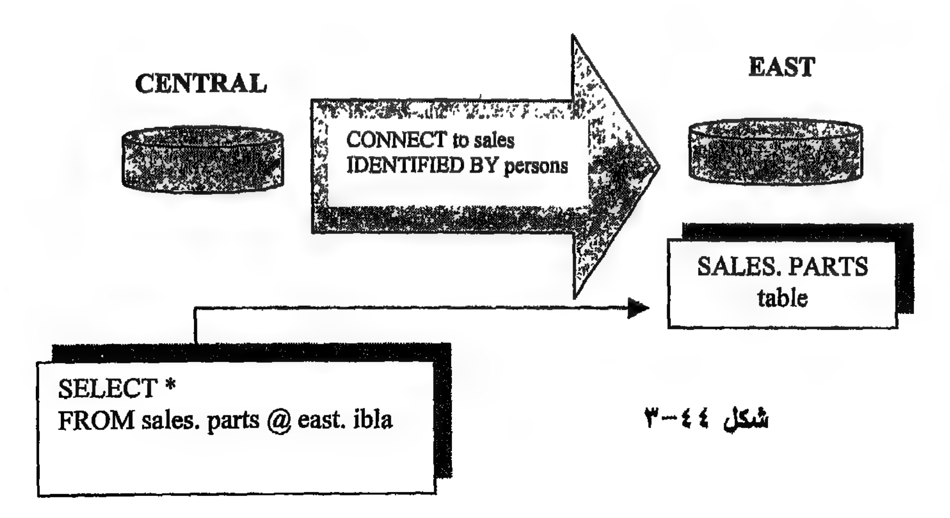
ارتباطات قاعدة المعطيات Database Links

حتى تستطيع الوصول عن بعد إلى قاعدة معطيات ضمن قواعد معطيات موزّعة يجب عليك تعريف ارتباطات قاعدة معطيات في قاعدة المعطيات المحلية الخاصة بك.

يحدد ارتباط قاعدة المعطيات ممر اتصال وحيد الاتجاه من قاعدة إلى أخرى.

يوضح الشكل ٤٤-٣ ارتباط قاعدة معطيات من قاعدة إلى أخرى في نظام قاعدة معطيات أوراكل الموزّعة.

ارتباط قاعدة المعطيات EAST. IBLA



شكل ٤٤-٣

يجب على ارتباط قاعدة المعطيات مطابقة اسم قاعدة المعطيات العام التي يدلّ عليها هذا الارتباط، مثلاً توضح التعليمة التالية كيفية إنشاء ارتباط قاعدة معطيات في قاعدة المعطيات المحلية CENTRAL يتم فيه تحديد الممر إلى قاعدة المعطيات EAST. IBLA:

CREATE DATABASE LINK east.ibla...

يمكن بعد ذلك للتطبيقات المتصلة بقاعدة المعطيات المحلية CENTRAL الوصول عن بعد إلى المعطيات في القاعدة EAST.IBLA.



يوضح المثال التالي كيفية استخدام تعليمة UPDATE لتعديل سطر في الجدول PARTS الموجود في قاعدة المعطيات EAST:

```
UPDATE sales.Parts @ east.ibla
SET unit_price = 100.50
WHERE id = 1;
```

أنواع ارتباطات قاعدة المعطيات

يمكنك عند إنشاء ارتباط قاعدة معطيات تحديد نوع اتصال محدد يستطيع مخدّم قاعدة المعطيات المحلية استخدامه لبدء دورة عن بعد في قاعدة المعطيات.

توجد ثلاثة أنواع لارتباطات قاعدة المعطيات ويحدد نوع الارتباط مستوى أمان خاص:

١-ارتباط قاعدة معطيات بمستخدم ثابت fixed user database link:

يحتوي هذا النوع من الارتباطات أثناء تعريفها على اسم مستخدم محدد وكلمة مرور.



يوضح المثال التالي كيفية إنشاء ارتباط قاعدة معطيات مستخدم ثابت لقاعدة معطيات EAST.IBLA وذلك عن بعد:

CREATE DATABASE LINK east.ibla
CONNECT sales IDENTIFIED BY person;

ويقوم هذا النوع من الارتباطات بتمرير معلومات الاتصال إلى المخدّم البعيد كنص كامل عند بدء دورة. لذلك يفضّل ترميز حزم اتصال المخدّم عند استخدام ارتباط قاعدة معطيات معطيات مستخدم ثابت.

٢- ارتباط قاعدة معطيات مستخدم متصل connected user database link:

وهو ارتباط لمستخدم متصل مع قاعدة المعطيات المحلية التي تقوم بتشغيل تطبيق. مثلاً إذا قمت بتشغيل تطبيق في قاعدة المعطيات المحلية الخاصة بي وأجريت الاتصال بالحساب NOUKARI، فإن المستخدم المتصل في هذا المفهوم هو NOUKARI.

٣- ارتباط قاعدة معطيات مستخدم حالي current user database link:

وهو سياق المستخدم الذي يتم فيه إجراء عملية ، فمثلا إذا قمت بالاتصال مع قاعدة المعطيات من خلال الحساب NOUKARI ونفّذت تطبيقاً يقوم بطلب الطريقة TOTAL_ ORDER الموجودة في نمط العنصر SALES.ORDER_ TYPE، فإنّ المستخدم الحالي الذي يقوم بتشغيل هذه الطريقة سيكون SALES وليس NOUKARI.

من الأنواع السابقة نلاحظ أنه قد يتم تضمين اسم المستخدم وكلمة المرور username/ password ضمن تعريف النوع الأول، بينما لا يتم ذلك في النوعين الآخرين.

كذلك فإنه يمكن استخدام ارتباطات قواعد معطيات المستخدمين المتصلين في أي توصيف لقاعدة معطيات موزّعة. بينما تسمح ارتباطات قواعد معطيات المستخدمين الحاليين بمضاعفة سياق الأمان security context من إجراء العمليات محلياً أو عن بعد.

مجالات إنشاء ارتباطات قواعد المعطيات

يسمح لك أوراكل بإنشاء ارتباطات قواعد معطيات ضمن ثلاث مجالات مختلفة ضمن نظام قواعد معطيات موزّعة وهي:

١- **ارتباط قاعدة معطيات خاصة Private database link**: وهو عبارة عن ارتباط قاعدة معطيات يتم إنشاؤه ضمن مخطط قاعدة معطيات خاص، ويمكن فقط لمالك ارتباط قاعدة المعطيات الخاص أو برامج PL/SQL الجزئية كالحزم البرمجية والإجرائيات والتوابع وغيرها التي تستخدم ارتباط قاعدة المعطيات الخاص بالوصول إلى المعطيات، أو العناصر الموجودة في قاعدة المعطيات البعيدة.

٢- **ارتباط قاعدة معطيات عام Public database link**: وهو عبارة عن ارتباط قاعدة معطيات يتم إنشاؤه في المجال PUBLIC ضمن قاعدة المعطيات. ويستطيع جميع المستخدمين والحزم البرمجية وغيرها من استخدام هذا الارتباط للوصول إلى المعطيات أو العناصر الموجودة في قاعدة المعطيات البعيدة.

٣- **ارتباط قاعدة معطيات شامل Global database link**: وهو ارتباط قاعدة معطيات يدار من قبل مخدّم أسماء أوراكل Oracle Names Server.

عندما تقوم بإنشاء مخدّم أسماء الشبكة Net٨، فإنه يقوم تلقائياً بإنشاء وإدارة ارتباطات قاعدة معطيات شاملة لكل قاعدة معطيات في الشبكة. ويمكن لجميع المستخدمين والبرامج في قاعدة المعطيات استخدام الارتباطات الشاملة للوصول إلى المعطيات وعناصر المخطط في قاعدة معطيات بعيدة، مما يسمح بإدارة هذه الارتباطات بشكل مركزي وبسيط.

قواعد المعطيات الموزّعة غير المتجانسة
Heterogeneous Distributed Database

في هذا النوع من قواعد المعطيات، يكون هنالك قاعدة معطيات واحدة على الأقل ليست قاعدة أوراكل.

يمكن استخدام مجموعة برمجيات إضافيّة هي Oracle Open Gateways من أجل دمج أنظمة أوراكل مع أنظمة أخرى سوية لإنشاء نظام قاعدة معطيات موزّعة.

التطبيقات وقواعد المعطيات الموزّعة Applications and Distributed Databases

عندما تقوم بتطوير التطبيقات للعمل في نظام قواعد معطيات موزّعة، يتوجب عليك فهم العديد من التقنيات.

سنقوم في الفقرات التالية بشرح كيفية تطوير التطبيقات لتتمكن من الوصول من بعد إلى المعطيات في قاعدة معطيات موزّعة باستخدام تعليمات SQL وبرامج PL/SQL.

الاستعلامات عن بعد *Remote Queries*

وهي عبارة عن تعليمة SELECT تقوم باستخراج المعلومات عن بعد من جدول أو أكثر موجودة على نفس العقدة البعيدة.



يوضح المثال التالي استعلاماً عن بعد يقوم باستخراج المعلومات من الجدولين CUSTOMERS, ORDERS الموجودين ضمن قاعدة المعطيات WEST:

```
SELECT o. id, c. company_ name
FROM sales. Orders @ west.ibla o,
Sales.Customers @ west.ibla c
WHERE o. cust_ id = c. id;
```

الاستعلامات الموزّعة *Distributed Queries*

تقوم الاستعلامات الموزّعة باستخراج المعلومات من قاعدتي معطيات أو أكثر مختلفة.

يوضح المثال التالي استعلاماً موزّعاً يدمج المعلومات بين الجدول المحلي ORDERS والجدول البعيد CUSTOMERS:

```
SELECT o.id, c.company_ name
FROM sales.Orders o, sales.Customers @ west.ibla c
WHERE o.cust_id = c.id;
```

التعديلات عن بعد Remote Updates

وهي تمكننا من إجراء تعديل المعطيات في جدول بعيد.



يوضح المثال التالي كيفيّة تعديل سطر في جدول PARTS ضمن قــاعدة المعطيات EAST:

```
UPDATE sales.Parts @ east.ibla
SET unite_price = 100.50
WHERE id = 1;
```

التعديلات الموزّعة Distributed Updates

يتم إجراء هذا النوع من التعديلات على مخدّمين أو أكثر باستخدام تعليمة واحدة.

الطريقة الوحيدة للقيام بذلك هي إنشاء إجرائية مخزنــة stored procedure أو طريقــة عنصر Object method تحتوي تعديلات بعيدة، كل تعديل منها خاص بقاعدة معطيــات مختلفة.



يوضح المثال التالي برنامج PL/SQL يقوم بإجراء موزّع:

```
BEGIN
UPDATE sales.Parts @ east.ibla
       SET ...;
UPDATE sales.items
       SET ...;
END;
```

طلبات الإجرائيات عن بعد Remote Procedure Calls

يمكن ضمن بنية قاعدة معطيات أوراكل الموزّعة طلــب تنفيــذ الإجرائيــات المحليــة أو الإجرائيات البعيدة لإنجاز عمل ما.



يوضح المثال التالي استعلام يقوم بطلب تنفيذ إجرائية عن بعد على النمــط :SALES. ORDER_TYPE

```
SELECT o.order_ total
FROM sales. Orders @ east. Ibla o
WHERE id = 1;
```

التحويلات البعيدة Remote transactions

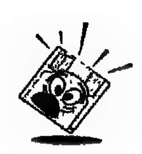
وهي عبارة عن تحويلات تحتوي على تعليمة بعيدة واحدة أو أكثر، وجميعها تدل على نفس قاعدة المعطيات البعيدة يوضح المثال التالي تحويلاً بعيداً يقوم بتعديل المعطيات في قاعدة المعطيات EAST فقط:



```
UPDATE sales.Parts @ east.ibla
       SET ...
       WHERE ... ;
UPDATE sales.Parts @ east.ibla
       SET ...
       WHERE ... ;
UPDATE sales.Parts @ east.ibla
       SET ...
       WHERE ... ;
COMMIT;
```

التحويلات الموزّعة Distributed transactions

وهي عبارة عن تحويلات تحتوي تعليمة واحدة أو أكثر تقوم بتعديل المعطيات في قواعد معطيات مختلفة.



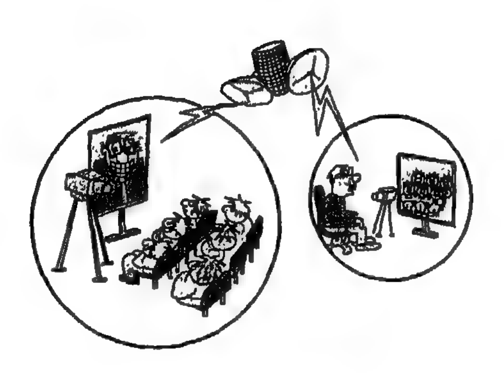
يوضح المثال التالي تحويلاً موزعاً يقوم بتعديل المعطيات في عدة قواعد معطيات:

```
UPDATE sales.Parts
  SET ...
  WHERE ... ;
UPDATE sales.Parts @ west.ibla
  SET ...
 WHERE ... ;
UPDATE sales.Parts @ west.ibla
   SET ...
   WHERE ... ;
COMMIT;
```

ارتباطات قاعدة المعطيات وقاموس المعطيات

يحتوي قاموس المعطيات على العديد من المشاهد الموافقة لارتباطات قاعدة المعطيـــات هي:

☆ المشـــاهد المتعلقـــة بارتباطـــات قـــاعدة المعطيـــات: DBA_DB_LINKS, ALL_DB_LINKS, USER_DB_LINKS.





أوراكل٨ وتناسخ المعطيات

Oracle8 and Data Replication

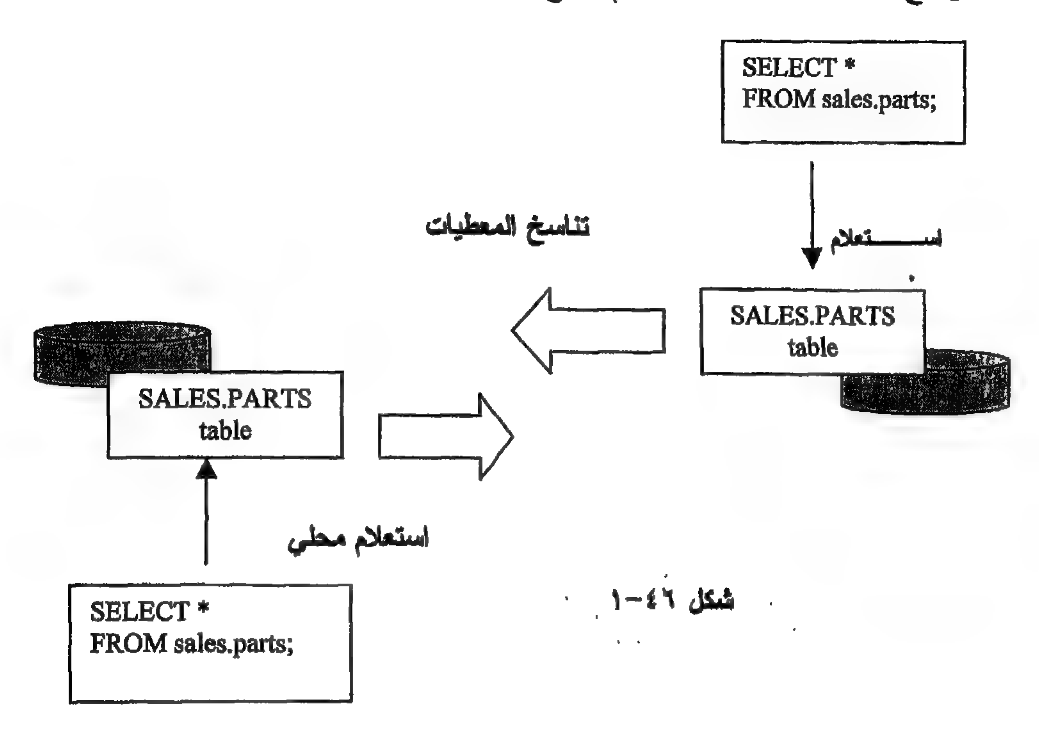
كما رأينا فإن نظم قواعد المعطيات الموزّعة هي أدوات فعالة يمكنك استخدامها لدمج مصادر المعطيات المنفصلة في قاعدة معطيات وحيدة وعامة لاستخدامها ضمن التطبيقات.

لكن هناك خصائص عامة لقواعد المعطيات الموزّعة تجعل منها غير مناسبة لبعض أنماط التطبيقات، فمثلاً لا يمكن لتطبيق ما استخدام قاعدة معطيات موزّعة وتحويلات موزّعة إلا إذا كانت جميع قواعد المعطيات في النظام متاحة بشكل متزامن وفي أغلب الأوقات.

يمكنك باستخدام تناسخ المعطيات دعم مختلف أنماط قواعد المعطيات الموزّعة ومختلف التطبيقات.

وتناسخ المعطيات هي إجراء نسخ وصيانة عناصر قاعدة المعطيات في مختلف قواعد المعطيات التي تشكل نظام قاعدة المعطيات الموزّعة.

يوضح الشكل ٤٦-١ أحد أنماط نظم تناسخ المعطيات:



شكل ٤٦-١

هنالك العديد من الفوائد التي يعطيها تناسخ المعطيات ولا يمكننا الحصول عليها في أنظمة قواعد المعطيات الموزّعة أهمها:

- ☆ يمكن لتناسخ المعطيات تحسين أداء تطبيق وأداء شبكة نظام قواعد المعطيات الموزّعة، والسبب في ذلك هو أنه يمكن للتطبيقات الوصول إلى معطيات النسخة الموجودة في قاعدة المعطيات المحلية بدلاً من الوصول إلى المعطيات في قواعد المعطيات البعيدة المتاحة ضمن الشبكة.
- ☆ يمكن لتناسخ المعطيات تحسين استخدام التطبيقات لأن تناسخ المعطيات يتواجد في عدة مواقع. فإذا أصبح موقع ما غير متاح لعطل في الشبكة أو النظام، فإنه يستطيع استخدام خيارات أخرى للوصول إلى المعطيات.

استخدامات تناسخ المعطيات Uses for Data Replication

يمكن لتناسخ المعطيات أن يكون مفيداً لأنماط عديدة من التطبيقات. توضح لك الأمثلة التالية متى يمكنك استخدام تناسخ المعطيات لدعم التطبيقات:

☆ تناسخ المعطيات مفيدة لتوزيع نسخ من المعلومات الهامة.

☆ يمكنك استخدام تناسخ المعطيات لنقل المعلومات من قاعدة معطيات إلى مخازن أخرى للمعطيات. يمكنك مثلاً استخدام ميزات نسخة معطيات أوراكل لنسخ المعطيات في أوقات محددة من قاعدة معطيات أوراكل إلى مخزن معطيات ما.

☆ يمكن أن يكون تناسخ المعطيات الطريقة الوحيدة التي يمكنك استخدامها لدعم تطبيقات معالجة التحويلات التي تعمل باستخدام أجزاء غير متصلة. فمثلاً عندما يكون مستخدمي نظام معالجة التحويلات متنقلين وغير متصلين بشبكة الشركة أغلب الأحيان، يمكنك استخدام تناسخ المعطيات لإجراء التزامن بين معطيات قواعد المعطيات الخاصة بكل منهم وبين قاعدة معطيات الشركة المركزية.

أنماط تناسخ المعطيات Types of Data Replication

يدعم Oracle8 نمطين من تناسخ المعطيات:

١- تناسخ المعطيات الأساسية basic replication:

حيث يسمح هذا النوع في الوصول للقراءة فقط إلى معطيات جدول يتوضّع على موقع أوّلي يسمى أحياناً بالموقع الرئيسي master site. ويمكن للتطبيقات الاستعلام عن المعطيات في نسخة المعطيات المحلية لتجنّب الوصول إليها عن طريق الشبكة.

٢- تناسخ المعطيات المتقدمة advanced replication:

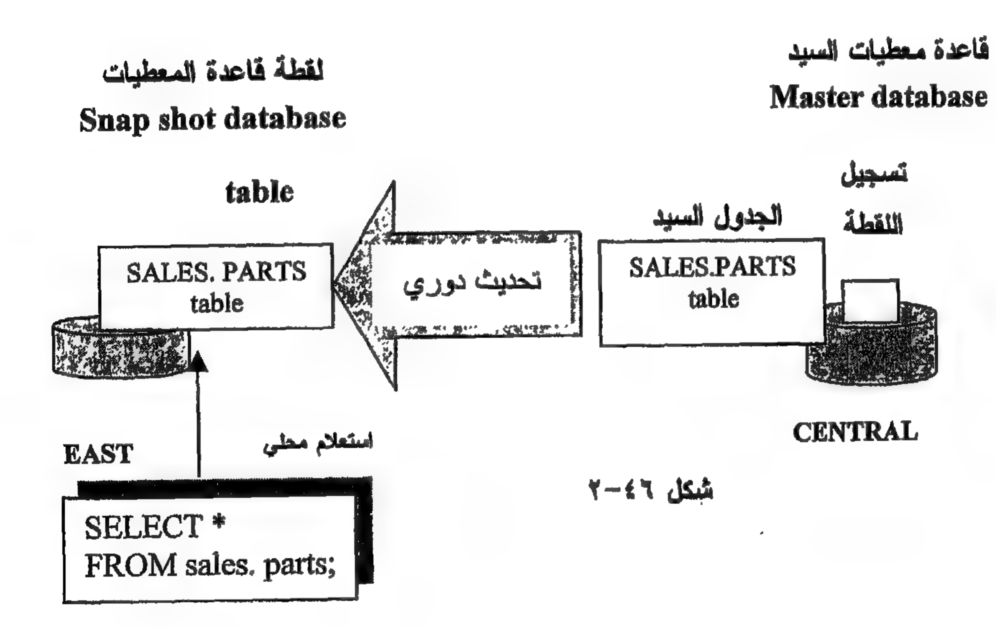
حيث يمكن للتطبيقات قراءة وتعديل نسخ الجداول الموجودة في النظام، لكن يجب إجراء توصيف خاص لكل مخدّم قاعدة معطيات يقوم بإدارة نسخ المعطيات في النظام.

تناسخ المعطيات الأساسية ولقطات القراءة فقط Basic Replication and Read Only Snapshot

يمكن باستخدام ميزات تناسخ المعطيات الأساسية دعم التطبيقات التي تحتاج للوصول لمعطيات جدول(للقراءة فقط) مصدّرة من موقع أوّلي. للقيام بذلك يمكنك إنشاء واستخدام لقطات جدول للقراءة فقط read_ only table snapshots، وهي عبارة عن نسخة محلية للجدول المصدّر من جدول رئيسي بعيد.

يمكن للتطبيقات الاستعلام عن المعطيات في لقطة جدول للقراءة فقط، لكن لا يمكنها إدراج أو حذف أو تعديل الأسطر فيها.

يوضح الشكل ٤٦-٢ بنية لقطة جدول للقراءة فقط في بنية نسخة معطيات أساسية.



شكل ٤٦-٢

تتشابه لقطة الجدول table snapshot مع المشهد View لأنك تقوم بتعريف بنية المعطيات المنطقية للقطة الجدول من خلال استعلام.



توضح تعليمة CREATE SNAPSHOT التالية كيفية إنشاء لقطة للجدول البعيد PARTS:

```
CREATE SNAPSHOT sales.parts AS
SELECT * FROM sales.parts @ central.ibla
```

تختلف لقطة الجدول table snapshot عن المشهد View في أنها تخزّن معطيات الجدول الذي يتم توليده من خلال تعريف الاستعلام.

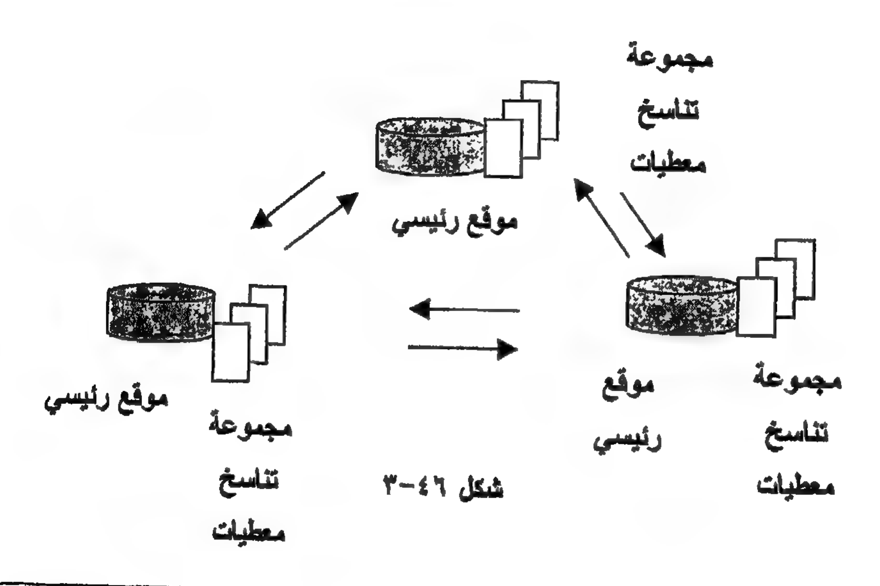
بيئة تناسخ المعطيات المتقدمة Advanced Replication Environment

من خلال هذه البيئة، يمكن قراءة وتعديل معطيات الجدول من خلال تناسخ المعطيات.

توجد طريقتان مختلفتان لتوصيف بيئة تناسخ المعطيات المتقدمة:

١- **تناسخ المعطيات لعدة مواقع الرئيسية Multimaster replication**: يسمح هذا النوع لعدة مواقع بإدارة المجموعات الكاملة من نسخ عناصر المخطط، ويمكن لها العمل متساوية حيث لا يوجد موقع يمكن اعتباره كموقع أولي. ويمكن للتطبيقات تعديل أية جداول منسوخة في أي موقع من المواقع الرئيسية.

يوضح الشكل ٤٦-٣ نظام نسخة معطيات بعدة مواقع رئيسية:

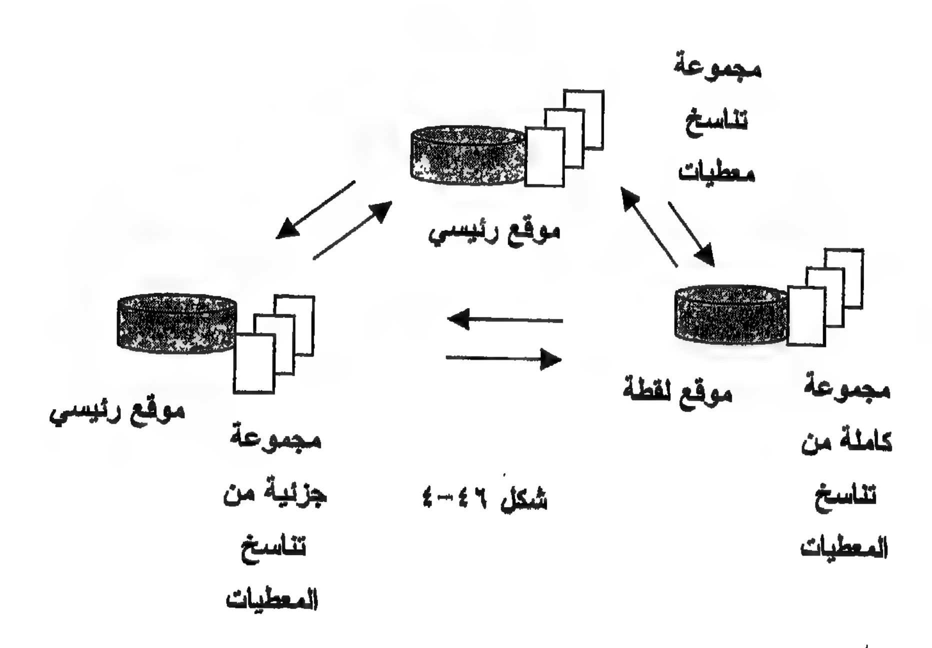


شكل ٤٦-٣

٢- مواقع اللقطات القابلة للتعديل **Updateable Snapshot sites**: عندما تقوم بإنشاء موقـــع رئيســـي واحد على الأقل في نظام نسخة معطيات متقدم، يمكنك عندئذ إنشاء مواقع لقطات مرتبطة بها مع لقطات قابلة للتعديل.

وتســـمح تســـهيلات نسخة معطيات أوراكل المتقدمة للتطبيقات بإدراج أو تعديل أو حذف الأسطر من اللقطات القابلة للتعديل.

يوضــــح الشـــكل ٤٦-٤ نظـــام نسخة معطيات متقدمة بموقع رئيسي واحد مع عدة مواقع لقطات قابلة للتعديل.



شكل ٤٦-٤

تشبه اللقطات القابلة للتعديل لقطات القراءة فقط إلى حد كبير، لكن تمتلك خصائص منفردة. مـــثلاً يقـــوم أوراكـــل بتحديث اللقطات القابلة للتعديل كجزء من مجموعة التحديث بشكل مطـــابق لـــلقطات القراءة فقط. إلا أنه يسمح لك أيضاً بتعديل المعطيات في اللقطات القابلة الـــتعديل، لذلك فإنه يجب على المخدّم الذي يقوم بإدارة موقع اللقطة أن يقوم بشكل نظامي بنشر التغييرات الحاصلة من اللقطات القابلة للتعديل إلى لقطة الجدول الرئيسي البعيد.





أوراكل٨ وقواعد المعطيات المتوازية

Oracle8 and Parallel Data Bases

سنقوم في هذا الفصل بشرح البنى المختلفة للحواسيب متعددة المعالجات multiprocessor computers وكيفيّة توصيف مخدّم أوراكل لهذه الأنظمة.

النقاط الأساسية التي سيتم شرحها في هذا الفصل:

* بنية أنظمة الحواسيب متعددة المعالجات كالذاكرة المشتركة shared memory والقرص المشترك shared disk والمعالجات المتعدّدة غير المشتركة shared nothing multiprocessor.
* كيفية توصيف نظام أوراكل لمعالجة الاستعلامات وتعليمات لغة معالجة المعطيات DML ضمن المعالجات المتوازية بحيث تستطيع الاستفادة بشفافية من إمكانيات المعالجة المتوازية.

* كيفية توصيف خيار مخدّم أوراكل المتوازي الخاص بالقرص المشترك وأنظمة الحواسيب متعددة المعالجات غير المشتركة، كذلك كيفية توصيف قاعدة المعطيات للوصول المتوازي إليها.

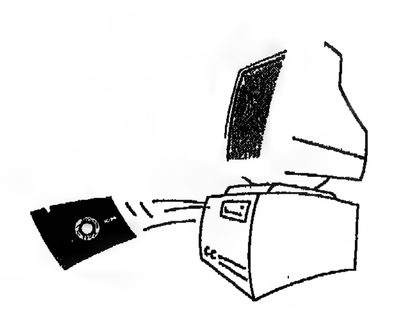
مقدمة عن المعالجة المتوازية

يمكن للعديد من التطبيقات التي تعمل باستخدام نظام حاسوبي بمعالج وحيد الاستفادة من بعض إمكانيات المعالجة المتوازية لأنظمة الحواسيب متعددة المعالجات.

فعندما يقوم تطبيق ما بإرسال طلب معين إلى حاسب وحيد المعالج سيقوم هذا المعالج بتنفيذ الطلب بنفسه، وتنتظر بقية التطبيقات أن ينتهي المعالج من تنفيذ هذا الطلب حتى تستطيع الحصول على طلباتها الخاصة.

أما الحواسيب متعدّدة المعالجات، فإنها تستخدم المعالجة المتوازية لتحسين أداء التطبيقات. فعندما يقوم تطبيق ما بإرسال طلبه إلى هذه الحواسيب، تقوم بتقسيم العمل إلى مهام جزئية منطقية ومن ثم معالجة المهام الجزئية بشكلٍ متوازٍ باستخدام المعالجات العديدة الموجودة ضمن النظام مما يقلّص زمن تنفيذ العمل إلى حدٍّ كبير.

نرمز لعدد المهام الفرعية الناتجة عن عمل منطقي واحد بدرجة التوازي degree of parallelism. ويتناسب تقليص زمن المعالجة الضروري لإنجاز مهمة ما بدرجة التوازي.

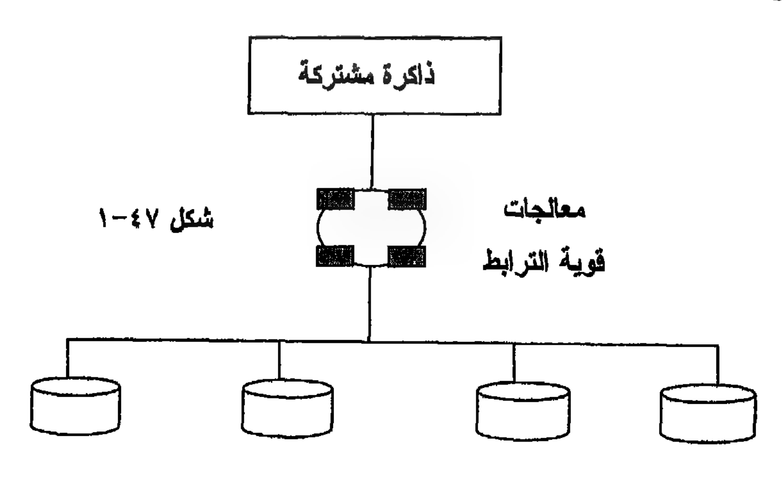


أنظمة الذاكرة المشتركة Shared Memory Systems

وهي أحد أنماط بنى الحواسيب متعددة المعالجات. وهي عبارة عن حاسوب متعدد المعالجات تتشارك بنفس الذاكرة ونفس قرص التخزين.

لهذا السبب تعرف هذه الأنظمة بالأنظمة قوية الترابط tightly coupled systems أو الأنظمة متعددة المعالجات المتناظرة Symmetric Multiprocessor Systems (SMP).

يوضح الشكل ٤٧-١ هذا النمط من الأنظمة:



شكل ٤٧-١

يقوم نظام التشغيل الخاص بأنظمة SMP بإدارة الوصول إلى الذاكرة المشتركة من قبل معالجات النظام بشكلٍ آمن. حيث يمنع نظام التشغيل أكثر من معالج واحد من الوصول إلى عنوان محدد في الذاكرة المشتركة.

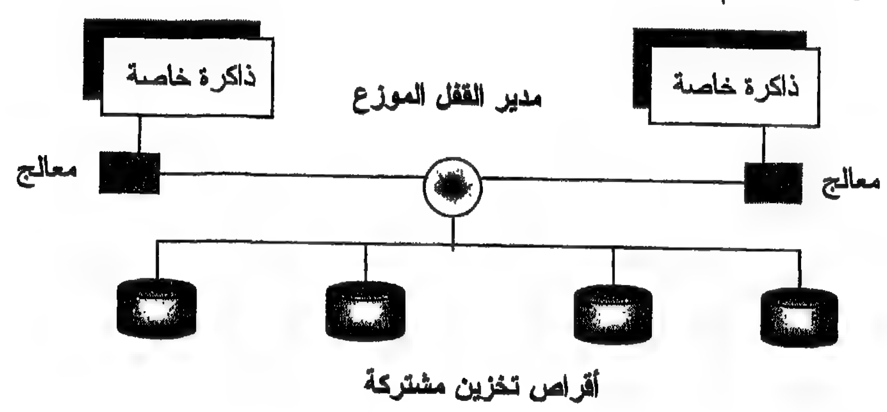
يوجد العديد من شركات الحواسيب التي تصنّع هذا النوع من الأنظمة بمعالجين أو أربعة أو ثمانية أو ستة عشر معالجاً على الأكثر.

أنظمة أقراص التخزين المشتركة Shared Disk Systems

وهو نمط آخر من أنماط الحواسيب متعددة المعالجات، حيث يمتلك كل معالج ذاكرته الخاصة لكنها تتشارك جميعاً بنفس قرص (أو أقراص) تخزين المعطيات.

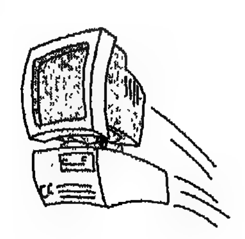
وكل معالج في هذه الأنظمة هو عبارة عن عقدة يتوجب عليها تنسيق الوصول إلى المعطيات المشتركة في القرص باستخدام مدير القفل الموزع المشترك Common distributed lock manager.

نسمي مجموعة العقد التي تتشارك بمجموعة أقراص التخزين بالتجمع Cluster. يوضح الشكل ٤٧-٢ نظام أقراص تخزين مشتركة:



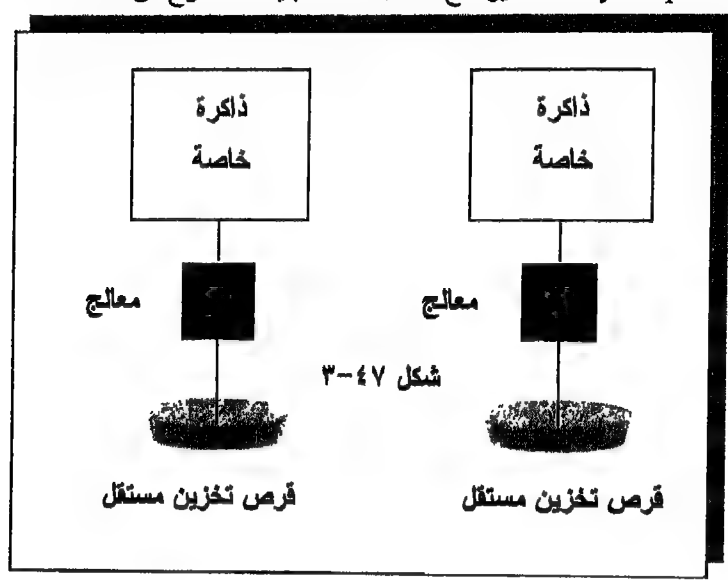
شكل ٤٧-٢

الميزة الهامة لهذه الأنظمة هي أنها تزودنا بإمكانية عالية لإتاحة الوصول إلى المعطيات الهامة، فعندما تتوقف عقدة بشكل كلي، يمكن للتطبيقات الاستمرار بالوصول إلى المعطيات بالاتصال مع أي عقدة أخرى ضمن التجمّع.



الأنظمة غير المشتركة Shared Nothing Systems

و هي عبارة عن أنظمة حواسيب بعدة معالجات تقوم بالعمل بشكل منفصل، لأن لكل معالج منطقة ذاكرة خاصة به وقرص تخزين خاص. لذلك فهي تعرف بالأنظمة ضعيفة الـترابط loosely coupled system، يوضح الشكل ٤٧-٣ بنية هذا النوع من الأنظمة :

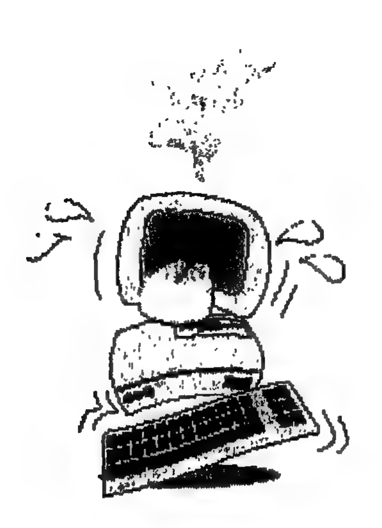


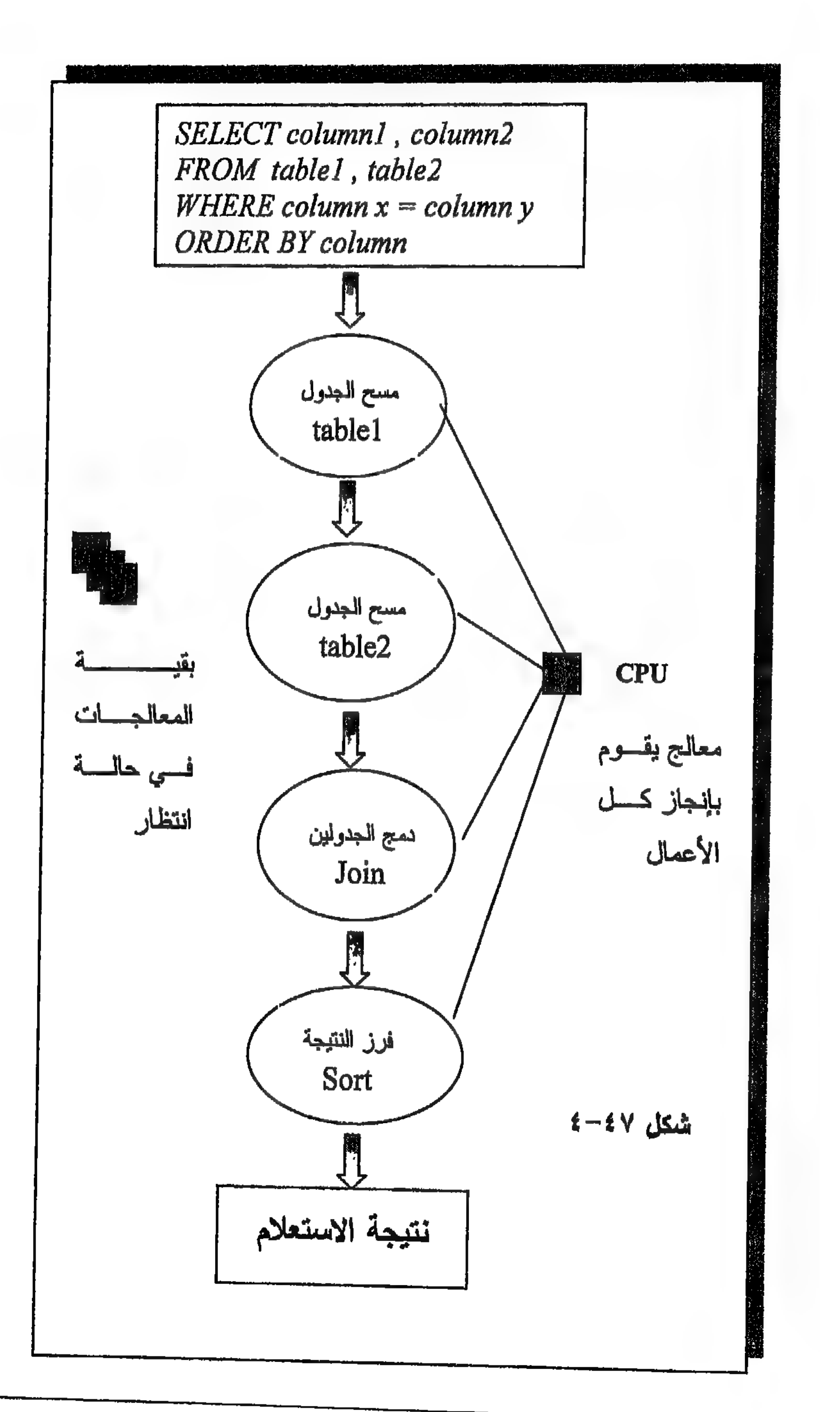
شكل ٤٧-٣

في هذا النوع من الأنظمة فإن العقدة هي عبارة عن معالج لها ذاكرته الخاصة و قرصـها الخاص، و يتم الاتصال بين العقد من خلال ممر داخلي عالي السرعة و لا توجد أي قيـود لعدد العقد في هذه الأنظمة.

معالجة تعليمات SQL بشكلٍ متوازٍ

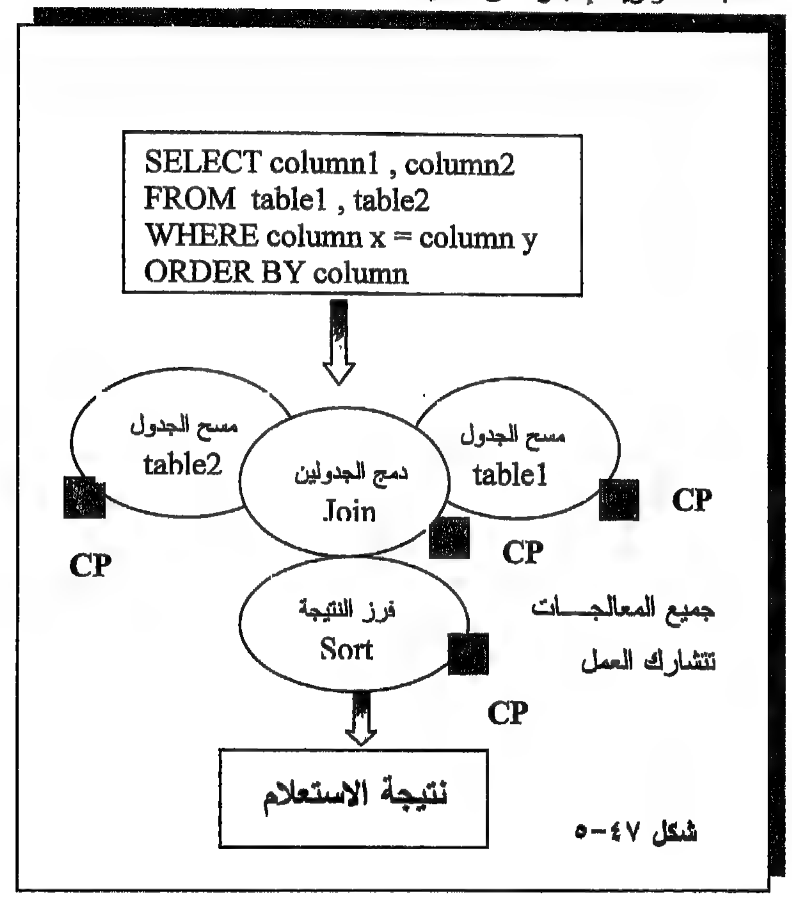
كما نعرف فإن الوظيفة الأساسية لأنظمة إدارة قواعد المعطيات كنظام أوراكل هي خدمة طلبات تطبيقات الزبون التي تقوم بالقراءة من قواعد المعطيات المشتركة أو الكتابة عليها.
ويمكن لمخدّم قاعدة معطيات أوراكل معالجة الطلبات على حاسوب متعدد المعالجات إما باستخدام المعالجة التسلسلية أو المعالجة التبرعية. فمثلاً إذا كانت لدينا التعليمة SQL تقوم بالدمج المفروز لجدولين، فإنه باستخدام المعالجة التسلسلية يقوم المخدّم بجلب أسطر الجدول الأول، ثم أسطر الجدول الثاني، بعدها يقوم بدمج هذين الجدولين ثم فرز النتيجة أخيراً يقوم بإرجاع هذه النتيجة إلى المستخدم. في حالة المعالجة التسلسلية إذاً يقوم المخدّم باستخدام معالج وحيد لإنجاز العمليات بشكل متسلسل، لذلك فإن الزمن اللازم لإنجاز استعلام هو مجموع أزمان الخطوات المطلوبة لإنجاز هذا الاستعلام. يوضح الشكل ٤٧-٤ كيفية استخدام المعالجة التسلسلية لإنجاز تعليمة SQL السابقة:




SELECT column1 , column2
FROM table1 , table2
WHERE column x = column y
ORDER BY column
مسح الجدول
table1
مسح الجدول
table2
دمج الجدولين
Join
فرز النتيجة
Sort
نتيجة الاستعلام
CPU
معالج يقوم بإنجاز كل الأعمال
بقية المعالجات في حالة انتظار

شكل ٧٤-٤

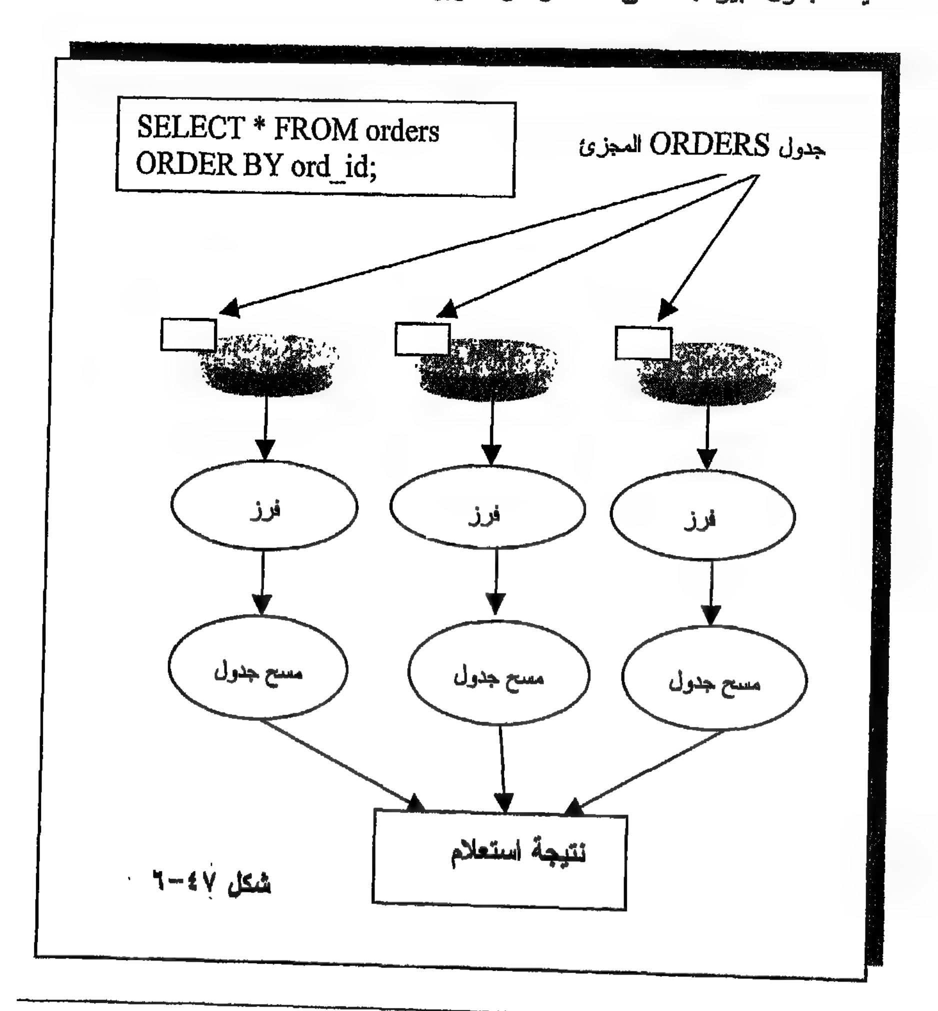
أما في حالة المعالجة المتوازية فإن النظام يقوم بتقسيم طلب قاعدة المعطيات إلـــى مــهام جزئية أصغر، ثم يقوم باستخدام جميع المعالجات المتاحة لمعالجة هذه المهام الجزئية بشكل متواز مما يساعد على تخفيض زمن المعالجة بشكل كبير. يوضح الشـــكل ٤٧-٥ كيفيـــة استخدام المعالجة المتوازية لإنجاز نفس التعليمة السابقة :



شكل ٤٧-٥

المعطيات المجزأة والمعالجة المتوازية لتعليمات SQL

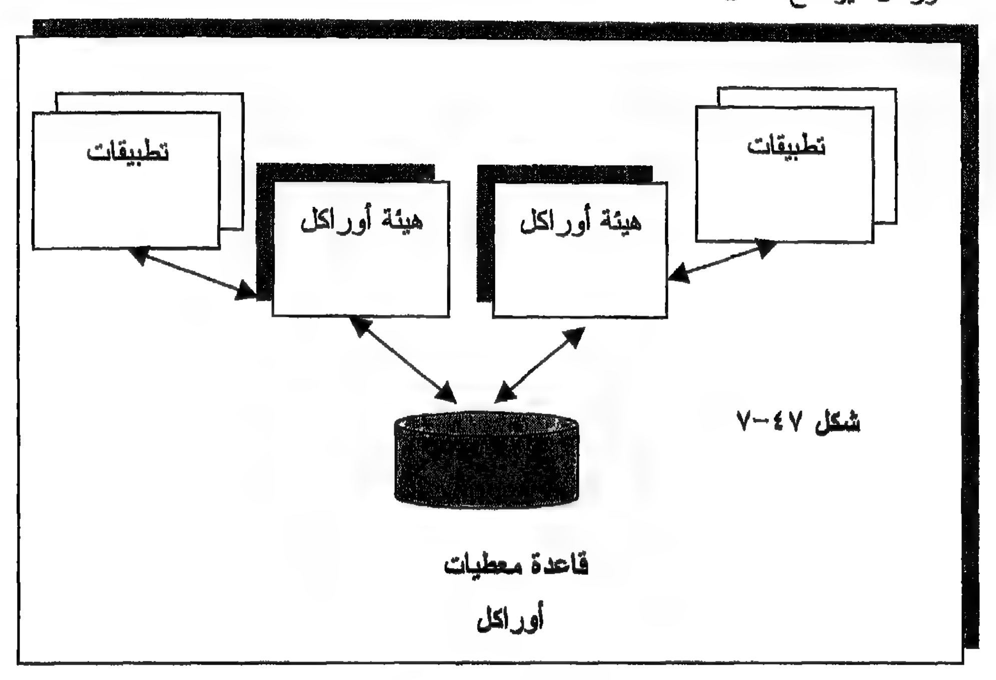
كما ذكرنا سابقاً فإن المعطيات المجزأة partitioned data عبارة عن مجموعة من المعطيات الموزّعة على عدة أقراص تخزين لحل مشاكل الاختناق الناتجة عن المجال المحدود للقراءة والكتابة من قرص وحيد. وحتى الاستعلامات المتوازية قد لا تكون قادرة على إنجاز عملياتها بشكلٍ جيد عندما تحاول الوصول إلى معطيات غير مجزأة. يوضح الشكل ٤٧-٦ كيف يمكن تنفيذ استعلام متوازي بشكل أسرع بكثير عندما يتم تجزيء معطيات جدول كبير جداً على عدة أقراص تخزين.



شكل ٤٧-٦

مخدّم أوراكل المتوازي Oracle Parallel Server

وهـو الـنمط التالي من أنماط المعالجة المتوازية والذي يمكّننا من الوصول المتوازي إلى قـاعدة المعطيـات حيـث يمكـن لعدّة هيئات الوصول المتزامن إلى نفس قاعدة معطيات أوراكل. يوضح الشكل ٧-٤٧ هذا النمط:



شكل ٧-٤٧

لكل هيئة في المخدّم المتوازي مجموعة خاصة من إجراءات الخلفية وذاكرة SGA، وتقوم بتركيب وفتح نفس قاعدة المعطيات في النمط المتوازي لإعطاء طرق مختلفة للتطبيقات من أجل القيام بالاتصال واستخدام قاعدة المعطيات المشتركة.


الباب الثالث عشر



٤٨. مساعد نشر وب.





مساعد نشر وب
Web Publication Assistant

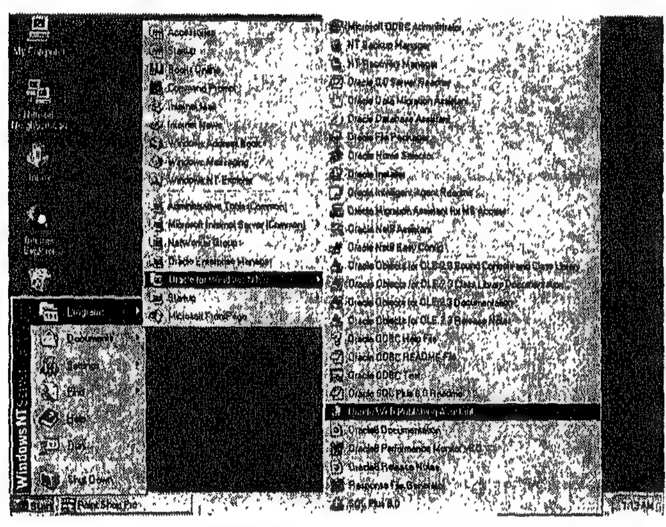
مساعد نشر وب Web publishing Assistant هو أحد المنتجات الجديدة مع Oracle ٨ ضـــمن نســـخة Windows NT والتي تساعد على إنشاء صفحات وب.

ويعتـــبر مـــن الأدوات السهلة الاستخدام والتي تساعد في إنشاء صفحات وب اعتماداً على معطيـــات قواعـــد معطيات أوراكل. هذه الصفحات هي صفحات ثابتة إلا أنّه يمكن إعادة إنشائها باستخدام قواعد نظامية، لذلك يمكن اعتبارها قابلة للتعديل.

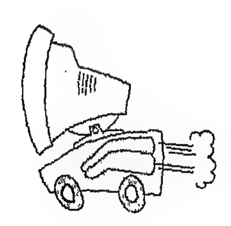
إنشاء صفحة وب ثابتة

اتبع الخطوات التي سنقوم بشرحها لاستخراج المعطيات من قاعدة معطيات أوراكل وإنشاء صفحة وب ثابتة. يمكنك ذلك من خلال معلومات صفحات وب الحالية دون الحاجة للوصول إلى هذه المعلومات من خلال قاعدة المعطيات.

١-قم بتشغيل Web Publishing Assistant من مجموعة برامج Oracle for Windows NT، انظر الشكل ٤٨-١.

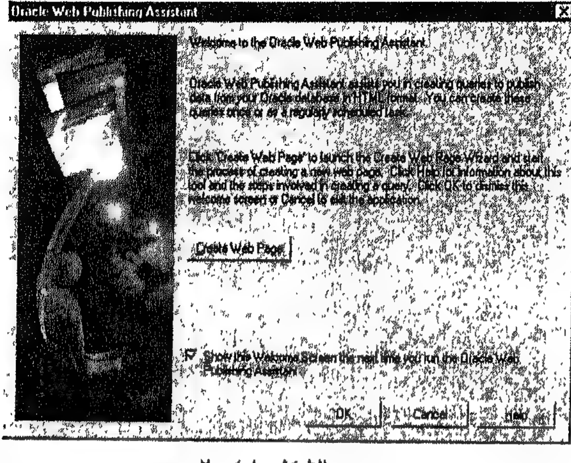


الشكل ٤٨-١



ستظهر نافذة الترحيب، انقر زر OK.

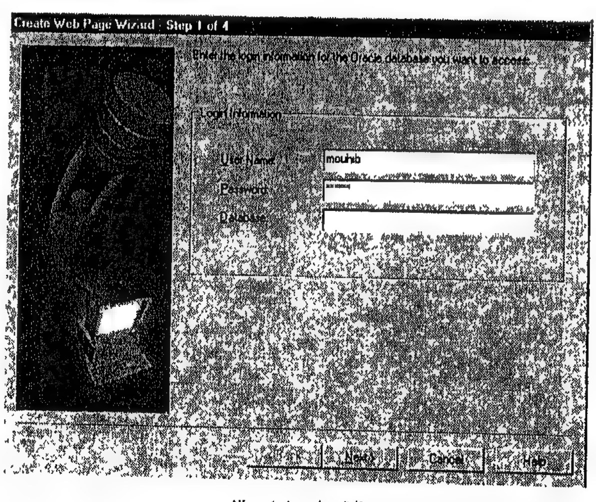
النافذة الثانية هي نافذة البرنامج الرئيسية (انظر الشكل ٤٨-٢). تسمح لك هذه النافذة بإظهار صفحات وب الفعالة التي قمت بتعريفها. وعلى اعتبار أنك لم تقم حتى الآن بتعريف أية صفحة فستظهر هذه النافذة خالية.



الشكل ٤٨-٢

الآن و حتى تقوم بإنشاء صفحة وب جديدة باستخدام هذا البرنامج، يمكنك استخدام معالج إنشاء صفحات وب Create Web Page Wizard لتشغيل هذا المعالج انقر زر Create Web Page.

ستظهر لك النافذة الأولى من نوافذ المعالج. تستخدم هذه النافذة لتعريف الاتصال مع قاعدة المعطيات التي سنحتاجها للحصول على المعطيات (انظر الشكل ٤٨-٣). تطلب منك هذه النافذة إدخال المعلومات التالية:



الشكل ٤٨-٣

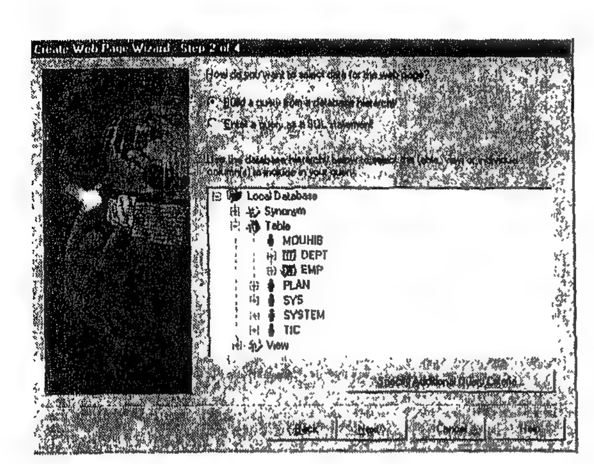
☆ اسم المستخدم User name.

☆ كلمة المرور Password.

☆ اسم قاعدة المعطيات Database.

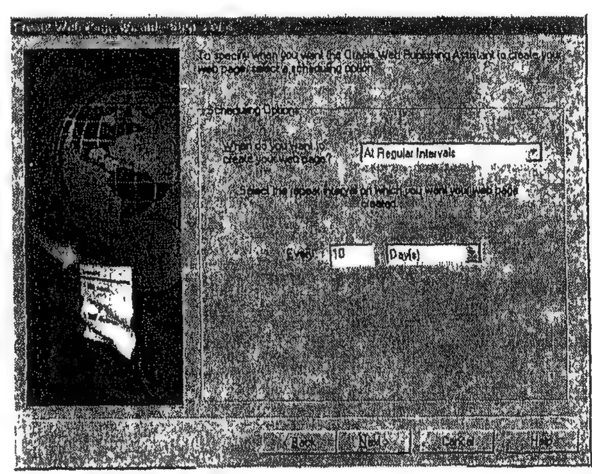
طبعاً هذا يسمح لعدة مستخدمين بإنشاء صفحات وب.

أما النافذة الثانية من نوافذ المعالج فتسمح لك بتحديد المعطيات التي سيتم إظهارها على صفحات وب. يمكن أن تكون هذه المعطيات على شكل جدول أو استعلام (انظر الشكل ٤٨-٤).



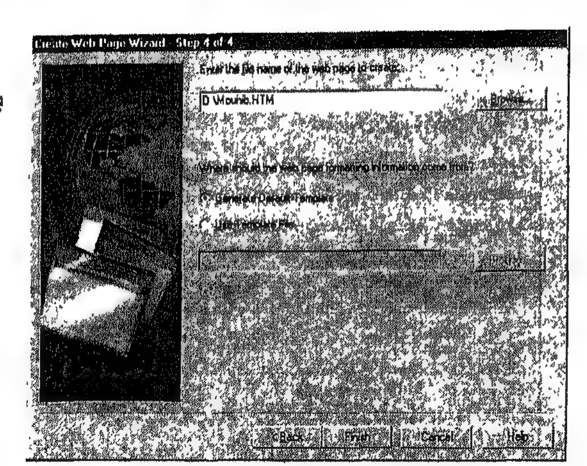
الشكل ٤٨-٤

النافذة الثالثة من نوافذ المعالج تستخدم لتعريف جدولة schedule التعديلات وذلك من أجل تحديث المعطيات الموجودة في قاعدة المعطيات اعتمادا على الجدولة الزمنية المحددة (انظر الشكل ٤٨-٥).



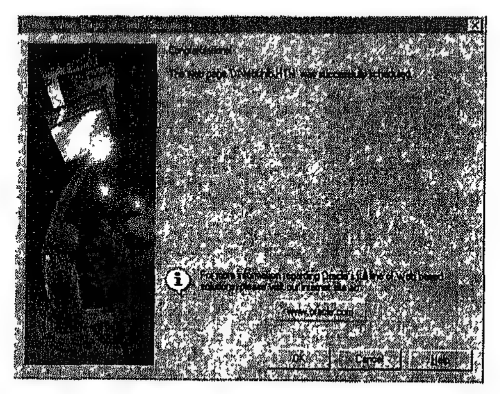
الشكل ٤٨-٥

أمـا النافذة الأخيرة فتسمح لك بـتعريف اسم صفحة وب التي قمـت بإنشـائها واختيار ملف قـالب معينّ أو اختيار القالب الافتراضي (انظر الشكل ٤٨-٦).



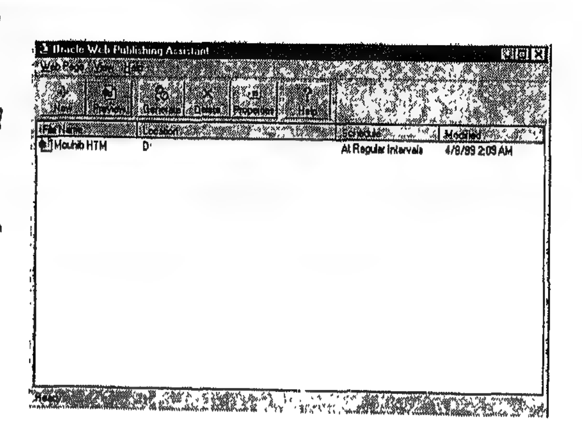
الشكل ٤٨-٦

تظهر بعد ذلك نافذة تلخيصيّة توضح إن كـان قـد تمّ توليد صفحة الوب المحـددة بنجاح أم لا (انظر الشكل ٤٨-٧).



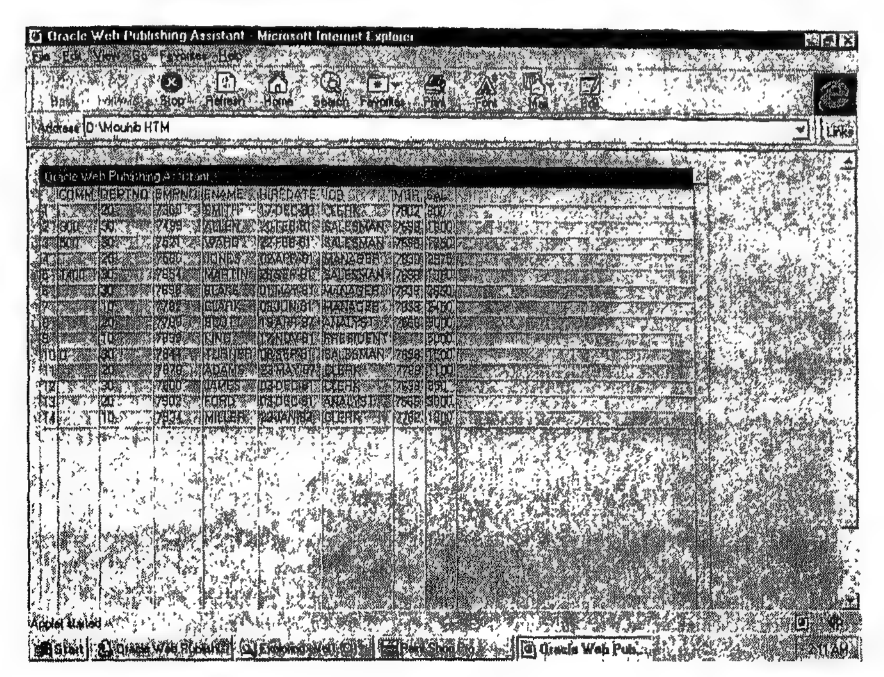
الشكل ٤٨-٧

تظهـر النافذة الرئيسية للبرنامج، لاحـظ ظهور صفحة وب الجديدة في هذه النافذة (انظر الشكل ٤٨-٨).

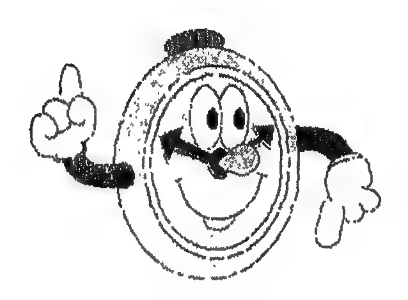


الشكل ٤٨-٨

يمكنك الآن توليد وإظهار صفحة وب التي قمت بإنشائها بالنقر على زر Generate، ثمّ Preview (انظر الشكل ٤٨-٩).



الشكل ٤٨-٩




الباب الرابع عشر



- ملحق١ : ملف توليد قاموس المعطيات.
- ملحق ٢ : مثال عن كيفيّة بناء قاعدة معطيـات جديدة.
- ملحق٣ : وسطاء التمهيد.
- ملحق٤ : امتيازات النظام.
- المراجع.
- المصطلحات.



ملفات توليد قاموس المعطيات
Scripts for Data Dictionary

يوضح هذا الملحق ملفات السرد المطلوبة لتوليد قاموس المعطيات، وهي تنفّذ تلقائياً عندما تقوم بإنشاء قاعدة المعطيات.

اسم ملف السرد	الخيارات المطلوبة	الوصف
CATALOG. SQL	None	يقوم بإنشاء قاموس المعطيات ومرادفات synonyms عامة للعديد من مشاهد المعاينة فيه وتمنح وصول عام PUBLIC للمرادفات، تستدعي أيضاً CATAUDIT.SQL، CATEXP.SQL و CATDR.SQL.
CATAUDIT.SQL	None	يقوم بإنشاء أثر مدقق ومشاهد معاينة لقاعدة المعطيات (يتم تشغيله تلقائياً من خلال CATALOG.SQL) ويمكن حذفه بتشغيل CATNAUD.SQL.
CATEXP.SQL	None	يقوم بإنشاء جداول لقاموس المعطيات خاصة بعمليات

		الاستيراد والتصدير Import/Export (تشغل تلقائياً من خلال CATALOG.SQL).
CATLDR.SQL	None	يقوم بإنشاء مشاهد معاينة لاستخدام SQL*Loader (يشغّل تلقائياً من خلال CATALOG.SQL).
CATPARR.SQL	Parallel Server	يقوم بإنشاء مشاهد معاينة لقاموس المعطيات لمعلومات المخدم المتوازي Parallel Server.
CATTRUST.SQL	Trusted ORACLE	تعرف بنى ضرورية لمخدم Trusted ORACLE Server.
CATPROC.SQL	Procedural	تشغل جميع ملفات السرد Script الضرورية أو المستخدمة ضمن خيار الإجرائيات Procedural وهي: CATPRC.SQL, CATSNAP.SQL, CATRPC.SQL, STANDARD.SQL, DBMSSTDX.SQL, PIPDL.SQL, PIDIAN.SQL, DIUTIL.SQL, PISTUB.SQL, DBMSUTIL.SQL, DBMSSNAP. SQL, DBMSLOCK.SQL, DBMSPIPE.SQL, DBMSALRT.SQL, DBMSOTPT.SQL, DBMSDESC. SQL.

CATPRC. SQL	Procedural	يقوم بإنشاء مشاهد معاينة في قاموس المعطيات للإجرائيات المخزنة والحزم البرمجية وإجراءات قاعدة المعطيات (يشغل تلقائياً من خلال CATPRC.SQL).
CATSNP. SQL	Procedural Distributed	تقوم بإنشاء بنى ضمن قاعدة المعطيات لتخزين الـ Snap Shots والمحافظة عليها (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ CATPRC.SQL).
CATRPC. SQL	Procedural Distributed	تقوم بإنشاء مشاهد معاينة ضمن قاعدة المعطيات لمعلومات قواعد المعطيات الموزعة (يشغل تلقائياً من خلال CATPRC.SQL ويحتاج لـ CATPRC.SQL).
STANDARD. SQL	Procedural	تقوم بإنشاء حزم برمجية PL/SQL للخيار الإجرائي (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ CATPRC.SQL).
DBMSSTDX. SQL	Procedural	يتضمن كتلاً توسعيةً للحزم البرمجية القياسية (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ STANDARD.SQL).

PIPDL. SQL	Procedural	يقوم بإنشاء حزم برمجية PL/SQL للخيار الإجرائي (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ DBMSSTDX.SQL).
PIDIAN. SQL	Procedural	يقوم بإنشاء حزم برمجية PL/SQL للخيار الإجرائي (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ DBMSSTDX.SQL).
DIUTIL. SQL	Procedural	يقوم بإنشاء حزم برمجية PL/SQLللخيار الإجرائي (يشغل تلقائياً من خلال CATAPROC.SQL ويحتاج لـ DBMSSTDX.SQL).
PISTUB . SQL	Procedural	يقوم بإنشاء حزم برمجية PL/SQL للخيار الإجرائي (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ DBMSSTDX.SQL).
DBMSUTIL. SQL	Procedural	يقوم بإنشاء حزم برمجية للخيار الإجرائي (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ DBMSSTDX.SQL).
DBMSSNAP. SQL	Procedural Distributed	يقوم بإنشاء إجرائيات لإدارة اللقطات Snapshotes (يشغل تلقائياً من خلال

		CATPROC.SQL ويحتاج لـ CATSNAP.SQL ويجب عليه تشغيله على اللقطة snapshot وعلى الجدول الرئيسي سوية).
DBMSALRT. SQL	Procedural	تسمح للمستخدمين وللتطبيقات باستخدام أحداث الانذارات (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL).
DBMSLOCK. SQL	Procedural	يسمح للمستخدمين وللتطبيقات باستخدام حزم القفل (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL).
DBMSMAIL. SQL	Procedural	يسمح للمستخدمين وللتطبيقات بإرسال رسائل Oracle*Mail (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL. ويجب عليه تشغيله في قاعدة المعطيات المرسلة وتشغيل UTLMAIL.SQL في قاعدة المعطيات المستقبلة).
DBMSOTPT. SQL	Procedural	يسمح لمطوري التطبيقات باستقبال I/ O من الإجرائيات. (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL).

DBMSPIPE. SQL	Procedural	يسمح للدورات في نفس المشتق بالاتصال مع بعضها. (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL).
DBMSDESC. SQL	Procedural	يسمح بإنشاء حزم برمجية تسمح لك بتوصيف المتغيرات والقيم الناتجة عن وحدات البرامج. (يشغل تلقائياً من خلال CATPROC.SQL ويحتاج لـ PISTUB.SQL).
CATALOGS. SQL	None	إنشاء مشاهد معاينة ضمن قاموس معطيات بالنسخة ٦
CATBLOCK. SQL	None	إنشاء مشهد المعاينة BLOCKING_LOCKS يظهر ما هي الأقفال locks التي توقف النظام.
CATDBSYN. SQL	None	إنشاء مرادفات خاصة لمشاهد _DBA في قاموس المعطيات.
CATEXP٦. SQL	None	إنشاء مشاهد معاينة لاستخدام النسخة ٦ من الأداة EXPORT ضمن ORACLE٧.
CATNOAUD. SQL	None	يقوم بحذف جميع آثار المدققات المنشأة بـ CATAUDIT.SQL والمتضمنة معطياتها ومشاهدها.
CATNOPRC. SQL	None	لحذف جميع البنى ضمن قاموس المعطيات والتي تم إنشاؤها

		باستخدام CATPRC.SQL.
CATSTAT. SQL	None	يقوم بتنفيذ تعليمة ANALYZE علـى بنى في قاموس المعطيات للسـماح باخـتزال أمثل لتعليمات SQL الداخلية.
UTLBSTAT. SQL	None	يقوم بعملية بدء تجميع إحصائيات عـن تحسـن الأداء (تنتهي بـ UTLESTAT.SQL).
UTLCHAIN. SQL	None	يقـوم بإنشاء جداول لتخزين خرج تعـليمة ANALYZE مع خيار CHAINED ROWS.
UTLDTREE. SQL	Procedural	يقوم بإنشاء جداول ومشاهد معاينة لإظهار الارتباطات بين العناصر.
ULESTAT. SQL	None	يقـوم بعمـلية إنهـاء تجميـع إحصائيات عن تحسن الأداء التي بدأت بـ UTLBSTAT.SQL.
UTLEXCPT. SQL	None	يقـوم بإنشاء الجدول الافتراضي EXCEPTIONS لـتخزين الاستثناءات عن المقيدات.
UTLEXCP ٦. SQL	None	يقوم بإرجاع قائمة بالعناصر التي لـم يتم تصديرها باستخدام النسخة ٦ من SQL * Net export.
UTLDIDXS. SQL	None	إظهـار نـتيجة تـنفيذ UTLIDXSS.SQL و UTLIDXSO.SQL.
UTLOIDSS. SQL	None	لتشـغيل UTLIDXSS.SQL على عدة أعمدة.

UTLLOCKT. SQL	None	لإظهار lock wait- for graph على شكل بنية شجرة.
UTLMAIL. SQL	None	تسمح لقاعدة معطيات ORACLE*Mail باستقبال رسائل من إجرائيات (يحتاج لتشغيل DBMSMAIL.SQL على قاعدة المعطيات التي سترسل الرسائل).
UTLI	None	إعطاء سماحية PUBLIC الوصول لجميع الجداول المستخدمة من قبل SQL*DBA Monitors.
UTLSAMPL. SQL	None	لإنشاء جداول بسيطة (مثل DEPT, EMP) ومستخدمين (مثل SCOTT).
UTLSIDXS. SQL	None	لحساب كيفية انتقاء الأعمدة واختبار فيما إذا كان الفهرس الخاص بالعمود مناسب.
UTLXPLAN. SQL	None	لإنشاء الجدول PLAN- TABLE والذي يحتوي على خرج من تعليمة EXPLAIN PLAN.



مثال عن كيفية بناء قاعدة معطيات جديدة
Example for creating a new data base

يوضح هذا الملحق الملفات التي يتمّ توليدها طلب تشغيل هذا البرنامج من خلال سطر الأوامر Command line ووفق الشكل:

EXP 80 username / password [options....]

دها عند بناء قاعدة معطيات جديدة باستخدام Oracle Database Assistant وفق احتياجات قاعدة المعطيات:

ملف وسطاء التمهيد initORCL

db_name = Tarb
db_files = 1020
control_files = ("D:\Tarbia_DB\ctl1ORCL.ora", "D:\Tarbia_DB\ctl2ORCL.ora")
db_file_multiblock_read_count = 32
db_block_buffers = 3200
shared_pool_size = 12000000
log_checkpoint_interval = 8000
processes = 200
dml_locks = 500
log_buffer = 163840
sequence_cache_entries = 100
sequence_cache_hash_buckets = 89
#audit_trail = true
#timed_statistics = true
background_dump_dest = D:\Tarbia_DB\trace
user_dump_dest = D:\Tarbia_DB\trace
db_block_size =2048
compatible = 8.0.5.0.0
sort_area_size = 65536

```
log_checkpoint_timeout = 0
remote_login_passwordfile = shared
max_dump_file_size = 10240
```

ملف دفعي لتنفيذ ملفات SQL الخاصة بتوليد القاعدة sqlORCL.bat

```
set ORACLE_SID=ORCL
D:\orant\bin\oradim80  -new -sid ORCL -intpwd tarbia -startmode auto
-pfile D:\Tarbia_DB\initORCL.ora
D:\orant\bin\oradim80  -startup -sid ORCL -starttype srvc,inst -usrpwd
tarbia -pfile D:\Tarbia_DB\initORCL.ora
D:\orant\bin\svrmgr30 @ORCLrun.sql
D:\orant\bin\svrmgr30 @ORCL1run.sql
```

ملف SQL يحتوي على تعليمات بناء القاعدة ORCLrun.sql

```
spool D:\orant\database\spoolmain
set echo on
connect INTERNAL/tarbia
startup nomount pfile=D:\Tarbia_DB\initORCL.ora
CREATE DATABASE Tarb
LOGFILE 'D:\Tarbia_DB\logORCL1.ora' SIZE 1024K,
   'D:\Tarbia_DB\logORCL2.ora' SIZE 1024K
MAXLOGFILES 32
MAXLOGMEMBERS 2
MAXLOGHISTORY 1
DATAFILE 'D:\Tarbia_DB\Sys1ORCL.ora' SIZE 50M
MAXDATAFILES 254
MAXINSTANCES 1
CHARACTER SET AR8ISO8859P6
NATIONAL CHARACTER SET AR8ISO8859P6;
spool off
```

ملف SQL يحتوي على تعليمات بناء الفضـــاءات الجدوليـــة ومقـــاطع التراجع ORCLrun1.sql

```
spool D:\orant\database\spoolmain
set echo on
connect INTERNAL/tarbia
ALTER DATABASE DATAFILE 'D:\Tarbia_DB\Sys1ORCL.ora'
AUTOEXTEND ON;
CREATE ROLLBACK SEGMENT SYSROL TABLESPACE "SYSTEM"
STORAGE (INITIAL 100K NEXT 100K);
ALTER ROLLBACK SEGMENT "SYSROL" ONLINE;
@D:\orant\Rdbms80\admin\catalog.sql;
@D:\orant\Rdbms80\admin\catproc.sql
@D:\orant\Rdbms80\admin\caths.sql

REM **************TABLESPACE FOR
ROLLBACK*****************
CREATE TABLESPACE RBS DATAFILE
'D:\Tarbia_DB\Rbs1ORCL.ora' SIZE 10M
DEFAULT STORAGE ( INITIAL 1024K NEXT 1024K MINEXTENTS 2
MAXEXTENTS 121 PCTINCREASE 0);
ALTER DATABASE DATAFILE 'D:\Tarbia_DB\Rbs1ORCL.ora'
AUTOEXTEND ON;

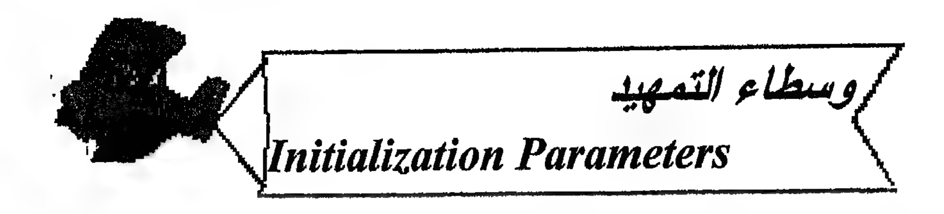
REM ***********Alter system tablespace ********************
ALTER TABLESPACE SYSTEM
DEFAULT STORAGE ( INITIAL 100K NEXT 100K MINEXTENTS 1
MAXEXTENTS 300 PCTINCREASE 1);

REM **************TABLESPACE FOR
USER*********************
CREATE TABLESPACE USR DATAFILE
'D:\Tarbia_DB\Usr1ORCL.ora' SIZE 3M
DEFAULT STORAGE ( INITIAL 50K NEXT 50K MINEXTENTS 1
MAXEXTENTS 121 PCTINCREASE 1);
ALTER DATABASE DATAFILE 'D:\Tarbia_DB\Usr1ORCL.ora'
AUTOEXTEND ON;
```

```
REM **************TABLESPACE FOR TEMPORARY*****************
CREATE TABLESPACE TEMPORARY DATAFILE 'D:\Tarbia_DB\Tmp1ORCL.ora' SIZE 10M
DEFAULT STORAGE ( INITIAL 100K NEXT 100K MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 0) TEMPORARY;
ALTER DATABASE DATAFILE 'D:\Tarbia_DB\Tmp1ORCL.ora' AUTOEXTEND ON;

REM **************TABLESPACE FOR INDEX*********************
CREATE TABLESPACE INDX DATAFILE 'D:\Tarbia_DB\Indx1ORCL.ora' SIZE 10M
DEFAULT STORAGE ( INITIAL 50K NEXT 50K MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 1);
ALTER DATABASE DATAFILE 'D:\Tarbia_DB\Indx1ORCL.ora' AUTOEXTEND ON;

REM **** Creating two rollback segments ****************
CREATE PUBLIC ROLLBACK SEGMENT RB0 TABLESPACE "RBS"
STORAGE ( INITIAL 50K NEXT 50K MINEXTENTS 2 MAXEXTENTS 121 );
CREATE PUBLIC ROLLBACK SEGMENT RB1 TABLESPACE "RBS"
STORAGE ( INITIAL 50K NEXT 50K MINEXTENTS 2 MAXEXTENTS 121 );
ALTER ROLLBACK SEGMENT "RB0" ONLINE;
ALTER ROLLBACK SEGMENT "RB1" ONLINE;
alter user sys temporary tablespace TEMPORARY;
alter user system default tablespace USR;
alter rollback segment "SYSROL" offline;
spool off
```



وسطاء التمهيد
Initialization Parameters

يوضح هذا الملحق قائمة بوسطاء التمهيد التي يمكن استخدامها في ملف الوسطاء Parameters File مع أنماط هذه الوسطاء وقيمها الافتراضية ووصف كل منها.

اسم الوسيط	نمطه	القيمة الافتراضية	وصف الوسيط
gc_rollback_segments	integer	20	عدد مقاطع التراجع (خاص بالمخدمات المتوازية).
gc_save_rollback_locks	integer	20	عدد كتل المقاطع الموزّعة على كل الهيئات (خاص بالمخدمات المتوازية).
gc_segments	integer	10	عدد مقاطع جميع الهيئات (خاص بالمخدمات المتوازية).
gc_tablespaces	integer	5	عدد الفضاءات الجدولية التي يمكن أن تكون فعّالة (خاص بالمخدمات المتوازية).
global_names	boolean	FALSE	التحقق من توافق أسماء ارتباطات قواعد المعطيات.
ifile	string	[NULL]	اسم الملف الذي يحتوي على وسطاء تمهيد أخرى.
inite_sql_files	string	OS/ DEP	ملفات بناء قاعدة المعطيات.
instance_number	integer	0	عدد الهيئات الخاصة بإدارة توسيعات العناصر.
log_archive_buffer_size archivage	integer	OS/ DEP	حجم كل دارئ أرشفة.
log_archive_buffers	integer	OS/ DEP	عدد دارئات الأرشفة.
log_archive_dest	string	OS/DEP	مسار الوصول لسواقة الأرشفة الخاصة بملفات الإرجاع.
log_archive_format	string	OS/DEP	شكل أسماء الملفات المؤرشفة (%t) و

			.(%s)
log_archive_start	boolean	FALSE	تفعيل أو عدم تفعيل الأرشفة التلقائية.
log_buffer	integer	OS/DEP	عدد الثمانيات المحجوزة في SGA لذواكر الإرجاع المؤقتة.
log_checkpoint_interval	integer	OS/DEP	المجال بين نقطتي تحقّق.
log_chekpoint_timeout	integer	0	الزمن بالثواني بين نقطتي تحقّق.
log_files	integer	255	العدد الأعظم لملفات الإرجاع التي يمكن فتحها معاً.
log_simultaneous_copies	integer	1	العدد الأعظم من نسخ LATCH في الذاكرة المؤقتة.
log_small_entry_max_size	integer	OS/DEP	الحجم التمهيدي للـ LATCH.
max_dump_file_size	integer	500	الحجم الأعظم مقدراً بالكتل لملفات الأثر.
max_enabled_roles	integer	20	العدد الأعظم للوظائف لكل مستخدم.
max_rollback_segment	integer	30	العدد الأعظم لمقاطع التراجع المستخدمة.
mts_dispatchers	integer	NULL	توصيف إجراء الموزّع dispatcher.
mts_listener_address	integer	NULL	توصيف إجراء المستمع listener.
mts_max_dispatchers	integer		العدد الأعظم لإجراءات الموزّع.
mts_max_servers	integer		العدد الأعظم لإجراءات المخدم الموزّعة.
mts_servers	integer		عدد إجراءات المخدّم المولّدة.
mts_service	string	NULL	اسم الخدمة المرتبطة بالموزّع.
nls_currency	string		رمز عملة البلد.
nls_date_format	string		التنسيق الإفتراضي للتواريخ.
nls_date_language	string		ترجمة أسماء الأشهر والأيام.
nls_iso_currency	string		رمز العملة ضمن ISO.
nls_language	string	OS/DEP	نمط اللغة.
nls_numeric_characters	string		الفواصل بين المئات والآلاف.
nls_sort	boolean	FALSE	سلسلة الفرز.
nls_territory	string	OS/DEP	نمط الأرض.
open_cursors	integer	50	العدد الأعظم للمؤشرات المفتوحة لكل إجراء مستخدم.

open_links	integer	4	العدد الأعظم للاتصالات بالقواعد البعيدة لكل إجراء مستخدم.
os_authent_prefix	string	OPS$	سابقة اسم المستخدم من أجل الاتصالات التلقائية.
os_roles	boolean	FALSE	الوظائف التي يديرها نظام التشغيل.
processes	integer	50	العدد الأعظم لإجراءات النظام المتصلة بالوقت نفسه..
resource_limit	boolean	FALSE	سماحية التشكيلات الجانبية.
rollback_segments	string	NULL	اسماء مقاطع التراجع الخاصة المتاحة للهيئة.
row_cache_cursors	integer	10	عدد المؤشرات التكرارية في الذاكرة المؤقتة.
row_locking	string	ALWAYS	نمط قفل الجداول والأسطر.
sequence_cache_entries	integer	10	عدد السلاسل الموجودة في SGA المخصصة للوصول المباشر.
sequence_cache_hash_buckets	integer	7	عدد الأماكن المحجوزة للوصول السريع للسلاسل.
serializable	boolean	FALSE	
sessions	integer	1.1*PROC	العدد الكلي للدورات.
shared_pool_size	integer	1.5M	حجم منطقة SQL المشتركة.
single_process	boolean	FALSE	محاكاة الآلة وحيدة المهام.
sort_area_retained_size	integer		الحجم الأمثل لمنطقة الفرز.
sort_area_size	integer	OS/DEP	الحجم الأعظم للذاكرة الحقيقية المخصصة للفرز.
sort_spacemap_size	integer	OS/DEP	الحجم بالثمانيات لخريطة الفرز ضمن المؤشرات.
sql_trace	boolean	FALSE	تفعيل أو إلغاء تفعيل الأثر للهيئة.
temporary_table_locks	integer		العدد الأعظم للجداول المؤقتة التي يمكن إنشاءها في المقاطع المؤقتة.
thread	integer		عدد تشعبات الإرجاع (خاص بالمخدمات المتوازية).

timed_statistics	boolean	FALSE	إحصائيات الزمن للأداة Server Manager.
transactions	integer	1,1*PROC	العدد الأعظم للتحويلات المتزامنة.
transaction_per_rollback_segment	integer	20	العدد الأعظم للتحويلات المتزامنة لكل مقطع تراجع.
user_dump_dest	string	OS/DEP	مسار الوصول لأثر إجراءات المستخدم.



يوضح هذا الملحق قائمة بامتيازات النظام مع العمليات المسموحة لكل امتياز:

امتياز النظام	العمليات المسموحة
ANALYSE	
ANALYSE ANY	تحليل أي جدول أو فهرس أو تجمّع في قاعدة المعطيات.
AUDIT	
AUDIT ANY	التحقق من أي مخطط عناصر في قاعدة المعطيات.
AUDIT SYSTEM	تأهيل أو عدم تأهيل تعليمة وامتياز خيارات التحقق.
CLUSTER	
CREATE CLUSTER	إنشاء تجمّع في المخطط الحالي.
CREATE ANY CLUSTER	إنشاء تجمّع في أي مخطط.
ALTER ANY CLUSTER	تعديل أيّ تجمّع في قاعدة المعطيات.
DROP ANY CLUSTER	حذف أي تجمّع في قاعدة المعطيات.
DATABASE	
ALTER DATABASE	تعديل قاعدة المعطيات.
DATABASE LINK	
CREATE DATABASE LINK	إنشاء ارتباطات قاعدة معطيات خاصة بالمخطط الحالي.
INDEX	
CREATE ANY INDEX	إنشاء فهرس في أي مخطط وعلى أي جدول.
ALTER ANY INDEX	تعديل أي فهرس في قاعدة المعطيات.
DROP ANY INDEX	حذف أي فهرس.

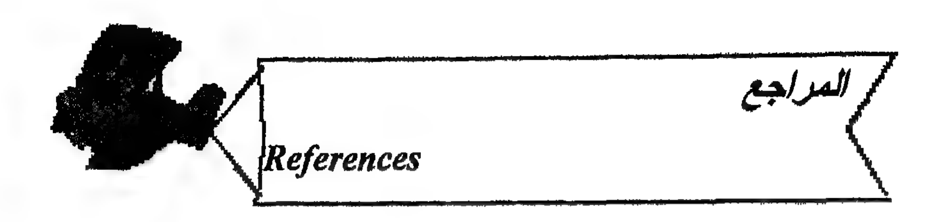
PRIVILEGE	
GRANT ANY PRIVILEGE	منح أي امتياز نظام (وليس امتياز عنصر).
PROCEDURE	
CREATE PROCEDURE	إنشاء إجرائيات مخزّنة أو دالات أو حزم برمجية في المخطط الحالي.
CREATE ANY PROCEDURE	إنشاء إجرائيات مخزّنة أو دالات أو حزم برمجية في أي المخطط (تحتاج أن يمتلك المستخدم الامتيازات التالية: ALTER ANY TABLE, BACKUP ANY TABLE, DROP ANY TABLE, LOCK ANY TABLE, COMMENT ANY TABLE, SELECT ANY TABLE, INSERT ANY TABLE, UPDATE ANY TABLE, DELETE ANY TABLE, GRANT ANY TABLE).
ALTER ANY PROCEDURE	ترجمة أي إجرائية مخزّنة أو دالّة أو حزمة برمجية في أي مخطط.
DROP ANY PROCEDURE	حذف أي إجرائية مخزّنة أو دالّة أو حزمة برمجية من أي مخطط.
EXECUTE ANY PROCEDURE	تنفيذ أي إجرائية مخزّنة أو دالّة أو حزمة برمجية أو إسناد أي متحول حزمة برمجية عامّ في أي مخطط.
PROFILE	
CREATE PROFILE	إنشاء تشكيل جانبي.
ALTER PROFILE	تعديل أي تشكيل جانبي في قاعدة المعطيات.
DROP PROFILE	حذف أي تشكيل جانبي من قاعدة المعطيات.
ALTER RESOURCE COST	تحديد قيم المصادر المستخدمة في جميع دورات المستخدم.
PUBLIC DATABASE LINK	
CREATE PUBLIC DATABASE LINK	إنشاء الارتباطات العامة لقاعدة المعطيات.
DROP PUBLIC DATABASE LINK	حذف الارتباطات العامة لقاعدة المعطيات.

PUBLIC SYNONYM	
CREATE PUBLIC SYNONYM	إنشاء مرادف عامّ.
DROP PUBLIC SYNONYM	حذف مرادف عامّ.
ROLE	
CREATE ROLE	إنشاء وظيفة.
ALTER ANY ROLE	تعديل أي وظيفة.
DROP ANY ROLE	حذف أي وظيفة.
GRANT ANY ROLE	منح أي وظيفة.
ROLLBACK SEGMENT	
CREATE ROLLBACK SEGMENT	إنشاء مقاطع تراجع.
ALTER ROLLBACK SEGMENT	تعديل مقاطع تراجع.
DROP ROLLBACK SEGMENT	حذف مقاطع تراجع.
SESSION	
CREATE SESSION	إنشاء دورة.
ALTER SESSION	حذف دورة.
RESTRICTED SESSION	إجراء الاتصال عندما تكون قاعدة المعطيات قد أقلعت باستخدام STARTUP RESTRICTED.
SEQUENCE	
CREATE SEQUENCE	إنشاء سلسلة في المخطط الحالي.
CREATE ANY SEQUENCE	إنشاء أي سلسلة في أي مخطط.
ALTER ANY SEQUENCE	تعديل أي سلسلة في أي مخطط.
DROP ANY SEQUENCE	حذف أي سلسلة في أي مخطط.
SELECT ANY SEQUENCE	إسناد أي سلسلة في أي مخطط.
SNAPSHOT	
CREATE SNAPSHOT	إنشاء لقطة في المخطط الحالي (يجب أن يمتلك المستخدم

	الامتياز CREATE TABLE).
CREATE ANY SNAPSHOT	إنشاء أي لقطة في أي مخطط (يجب أن يمتلك المستخدم الامتياز CREATE ANY TABLE).
ALTER ANY SNAPSHOT	تعديل أي لقطة في أي مخطط.
DROP ANY SNAPSHOT	حذف أي لقطة في أي مخطط.
SYNONYM	
CREATE SYNONYM	إنشاء مرادف في المخطط الحالي.
CREATE ANY SYNONYM	إنشاء أي مرادف في أي مخطط.
DROP ANY SYNONYM	حذف أي مرادف في أي مخطط.
SYSTEM	
ALTER SYSTEM	امتياز تعديل النظام.
TABLE	
CREATE TABLE	إنشاء الجداول في المخطط الحالي. تسمح أيضاً بإنشاء الفهارس (متضمنة قيود التكامل) على الجداول في المخطط الحالي (يجب أن يمتلك المستخدم الممنوح نصيب في الفضاء الجدولي أو الامتياز UNLIMITED TABLESPACE).
CREATE ANY TABLE	إنشاء أي جدول في أي مخطط.
ALTER ANY TABLE	تعديل أي جدول في أي مخطط وترجمة أي مشهد في أي مخطط.
BACKUP ANY TABLE	إنجاز تصدير متزايد باستخدام الأداة Export.
DROP ANY TABLE	حذف أي جدول في أي مخطط.
LOCK ANY TABLE	قفل أي جدول أو أي مشهد في أي مخطط.
COMMENT ANY TABLE	إنشاء تعليق على أي جدول أو مشهد أو عمود في مخطط.
SELECT ANY TABLE	الاستفسار على أي جدول أو مشهد أو لقطة في مخطط.
INSERT ANY TABLE	إدراج أسطر في أي جدول أو مشهد في أي مخطط.
UPDATE ANY TABLE	تعديل أسطر في أي جدول أو مشهد في أي مخطط.

DELETE ANY TABLE	حذف أسطر من أي جدول أو مشهد في أي مخطط.
TABLESPACE	
CREATE TABLESPACE	إنشاء فضاءات جدولية.
ALTER TABLESPACE	تعديل فضاءات جدولية.
MANAGE TABLESPACE	جعل أي فضاء جولي غير فعّال أو جعل أي فضاء جدولي فعّال أو بدء وإنهاء النسخ الاحتياطي لأي فضاء جدولي.
DROP TABLESPACE	حذف فضاءات جدولية.
UNLIMITED TABLESPACE	استخدام مساحة محددة من أي فضاء جدولي.
TRANSACTION	
FORCE TRANSACTION	إجبار عملية التثبيت أو التراجع للتحويلات الموزّعة الحالية في قاعدة المعطيات المحلية.
FORCE ANY TRANSACTION	إجبار عملية التثبيت أو التراجع لجميع التحويلات الموزّعة في قاعدة المعطيات المحلية.
TRIGGER	
CREATE TRIGGER	إنشاء قادح في المخطط الحالي.
CREATE ANY TRIGGER	إنشاء أي قادح (في أي مخطط) مرتبط مع أي جدول في أي مخطط.
ALTER ANY TRIGGER	تأهيل أو عدم تأهيل أو ترجمة أي قادح في أي مخطط.
DROP ANY TRIGGER	حذف أي قادح في أي مخطط.
USER	
CREATE USER	إنشاء مستخدمين ومنحهم مساحات في أي فضاءات جدولية، كذلك تحديد الفضاءات الجدولية الافتراضية والمؤقتة، وتحديد التشكيل الجانبي.
BECOME USER	تحويل لمستخدم آخر (مطلوبة من أي مستخدم يقوم بإجراء استيراد كامل لقاعدة المعطيات).
ALTER USER	تعديل المستخدمين الآخرين كذلك تعديل كلمة مرور

	المستخدمين أو طريقة الاستيقان، وتحديد نصيبه في الفضاء الجدولي، أيضاً تحديد فضاءه الجدولي الافتراضي والمؤقت، وأخيراً تحديد التشكيلات الجانبية والوظائف الافتراضية.
DROP USER	حذف مستخدم آخر.
VIEW	
CREATE VIEW	إنشاء مشهد في المخطط الحالي.
CREATE ANY VIEW	إنشاء مشهد في أي مخطط (يحتاج أن يمتلك المستخدم الامتياز ALTER ANY TABLE, BACKUP ANY TABLE, DROP ANY TABLE, LOCK ANY TABLE, COMMENT ANY TABLE, SELECT ANY TABLE, INSERT ANY TABLE, UPDATE ANY TABLE, DELETE ANY TABLE, GRANT ANY TABLE).
DROP ANY VIEW	حذف أي مشهد في أي مخطط.



***ORACLE, A Beginner's Guide**, Michael Abbey-Michael I. Corey, Oracle Press, 1995.*

***ORACLE8 Architecture**, Steve Bobrowski, Foreword By Ken Jacobs, Oracle Press, 1998.*

***Teach Yourself ORACLE8 in 21 days**, Edward Whalen-Steve Adrien DeLuca, SAMS Publishing, 1998.*

***ORACLE7 Server Concepts Manual**, Steven Bobrowski, 1992.*

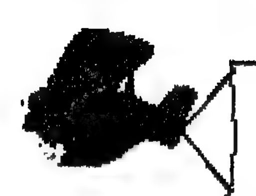
***ORACLE7 Server Administrators Guide**, Steven Bobrowski, 1992.*

***Oracle7 Server: Database Administration**, Volume1-2-3 Student Guide, Oracle Press, 1996.*

***Introduction to Oracle: SQL and PL/SQL**, Volume 1-2-3 Student Guide, Oracle Press, 1998.*

***Administration Oracle7**, G.Mopolo Moke, 1997.*





المصطلحات

إجراء بعيد Remote Process
إجراء مخدَم Server Process
إجراء مستخدم User Process
إجرائية مخزنة Stored Procedure
إحباط Abort
إخفاق Failure
أداء Performance
إدراج Insert
إرجاع Redo Log
أرشفة Archive
إسناد Reference
إلغاء منح Revoke
إنشاء Create
اجتزائية Modularity
اختزال Optimization
اختيار Select
ارتباط قاعدة معطيات Database Link
استثناء Exception
استخراج Retrieve
استرجاع Restore
استرداد Recovery
استعلام Query
استيراد Import
استيقان Authentication
امتداد أمامي Roll-Forward
امتداد خلفي Roll-Back
امتياز Privilege
بشفافية Transparently
بنية Architecture
بنيوية Structured
تثبيت Commit
تجمّع Cluster
تحديث Refresh
تحطّم Crash
تدقيق Audit
تحويل تسجيل Log Switch
تحويلات Transaction
تحويلات متزامنة Concurrent Transaction
تخويل Authorization
تراجع Rollback

ترجمة Compilation
ترحيل Migration
ترويسة Header
تسجيل Log
تسجيل الدخول Log In
تشعّب Threaded
تشفير/شيفرة Code
تشكيل جانبي Profile
تصدير Export
تضمين Encapsulate
تعديل Update
تعديل Alter
تعليق Comment
تقنية Mechanism
تكامل مرجعي Referential Integrity
تناسخ Replication
تنقيب عن معطيات Data Mining
جدول Table
جدولة Schedule
حالة الأحرف Case
حدث Event
حذف Delete
حذف Drop
حوض المشاركة Shared Pool
خرطوشة Cartridge
دارئ/ذاكرة مؤقتة Buffer
دالة Function

دقيقة/خالية من الأخطاء Accurate
دورة Session
دوري Periodic
ذاكرة حلقية Core
رئيسي Master
رديف Alias
رزمة/حزمة Package
سابقة Prefix
سلسلة Sequence
سلسلة Chaining
سياق Context
سياق الأمان Security Context
شحن المعطيات Load
شروط التكامل Integrity Constraints
صفات/واصفات Attributes
صفحة عمل Spreadsheet
طريقة Method
عام Public
عرقلة متبادلة Deadlock
عطب Crash
علّة Bug
عملية Operation
عنصر Object
غير فعّال Offline
غير متجانس Heterogeneous
فضاء جدولي Tablespace
فعّال Online

فهرس Index
قادح/منشط Trigger
قاعدة معطيات Database
قفل Lock
قيود Constraints
كونسول Console
لغة الاستعلام Query Language
لغة تعريف المعطيات Data Definition Language
لقطة Snapshot
مؤسسات افتراضية لحظية Instant Virtual Enterprise
مؤشر Cursor
متحول Variable
متعدد المعالجات Multiprocesser
متعدد الوسائط Multimedia
متناسقة Consistent
مجلد Catalog
محلّل Optimizer
مخازن Repositories
مخطط Schema
مداخل الإرجاع Redo Entries
مداخل التحويلات Transaction Entries
مدى Extent
مرآة القرص Disk Mirroring
مرادف Synonyrn
مساحة فارغة Free Space

مستعد للعمل Standby
مستمع Listener
مشارك/مشترك Shared
مشهد/منظار View
مضاعفة Mirror
معالج Wizard
معامل Operand
معامل Operator
مقطع Segment
مقيدات المصادر Resource Limits
مكرّس Dedicated
ملف إرجاع Redo Log File
ملف إرجاع مؤرشف Archive Log File
ملف تحكّم Control File
ملف معطيات Data File
منح Grant
موزّع/برنامج انتقاء Dispatcher
ميزة/طريقة Feature
نسخ احتياطي Backup
نصيب Quota
نظم قابلة للتدرج Scaleable Systems
نظم موزعة Distributed Systems
نقطة تدقيق Check Point
نموذج علاقاتي Relational Model
هيئة/حالة Instance

وثائق مترابطة Hypertext Documents

وزن Weight

وسطاء التمهيد Initialization Parameters

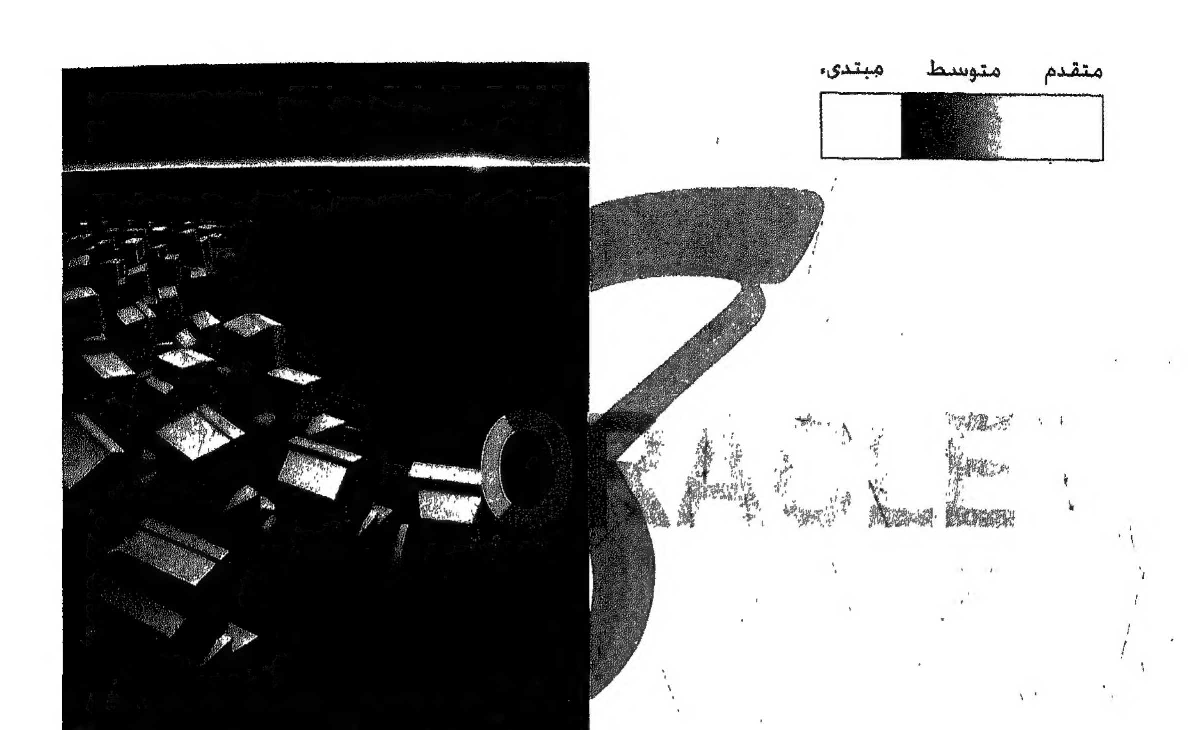
وصول متزامن Concurrent Access

وظيفة Role

وكيل Agent



متقدم متوسط مبتدىء



يعرض هذا الكتاب

المفاهيم الأساسية لنظام إدارة قواعد المعطيات العلائقية ORACLE وتتضمن المهام والمسؤوليات المطلوبة من مدير قاعدة المعطيات

- لغة الاستعلامات SQL ، لغة برمجة أوراكل PL/SQL
- توسيع ORACLE باتجاه البرمجة غرضية التوجه
- البنية الفيزيائية والمنطقية لقاعدة معطيات ORACLE
- العمليات الرئيسية الممكنة على قاعدة المعطيات
- حماية وأمان المعطيات
- بناء صفحات WEB باستخدام قاعدة معطيات ORACLE





سلسلة الرضا للمعلومات
دار الرضا للنشر